RH11-RS03

MAINT MODE DIAGNOSTIC

MD-11-DZRSE-C

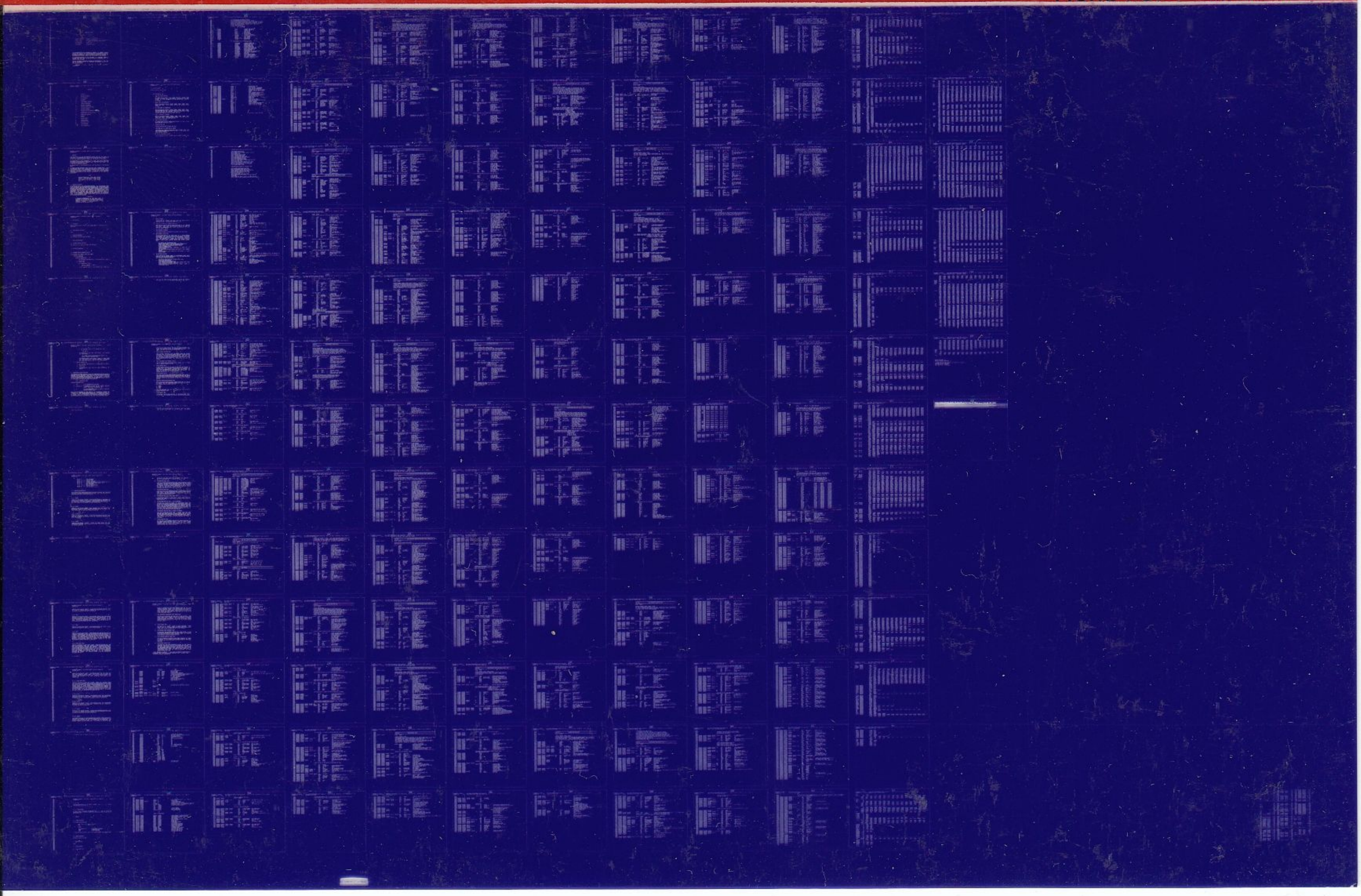
EP-DZRSE-C-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



```
                        .REM    %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRSE-C-D
PRODUCT NAME:	RH11-RS03 MAINTENANCE MODE DIAGNOSTIC
DATE CREATED:	AUGUST 1976
MAINTAINER:	DIAGNOSTIC GROUP

CONTENTS

1. ABSTRACT

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

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

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

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

NOTE

THIS DIAGNOSTIC WILL ALSO TEST RS03LA DRIVES. THEY MUST BE CONVERTED TO LOOK LIKE A RS03 TO BE TESTED. TURN DRIVES OFF LINE IF YOU DO NOT WISH TO TEST THEM.

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.

DESCRIPTION

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 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), IF SWITCHLESS CPU, SIMPLY

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

```
207                                     RS03 DRIVES ON THE SYSTEM
```

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

TYPE UNIT #

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

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

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

2. STARTING ADDRESS 220

A. SET SWITCHES (SEE SECTION 5), IF SWITCHLESS CPU SIMPLY

B. PRESS START

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

5. OPERATIONAL SWITCH SETTINGS

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

(I.E.) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE.

2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE; LAST DIGIT FOLLOWED BY <CR>.

3. ↑U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

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

```
   264                          SWITCH SETTINGS ARE:
```

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

5.1 SUBROUTINE ABSTRACTS

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

5.1.1 CLRDK

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

5.1.2 MRDMD

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

5.1.3 MRINT

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

5.1.4 MRIND

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

```
321
322                             5.1.5  MRCLK
```

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

5.1.6 MRCK

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

5.1.7 DSCK

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

5.1.8 XBIT

TRAPS TO A TAG CALLED ".XBIT". THIS ROUTINE GETS ONE DATA BIT THAT IS CURRENTLY BEING WRITTEN FROM THE DATA BUFFER IN CORE AND STORES IT IN A LOCATION CALLED NOWOD. THE PREVIOUS CONTENTS OF NOWOD IS STORED IN LASTOD. THIS INFORMATION IS USED BY THE CLKD1 AND CLKD0 ROUTINES TO DETERMINE THE CORRECT STATE OF THE MWDB (BIT 12) BIT IN RSMR WHEN WRITING. THIS ROUTINE MAKES BITS 16 AND 17 OF EACH DATA WORD (RS04 WRITES 18 BIT WORDS) EQUAL ZERO. THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE XBIT INSTRUCTION.

5.1.9 CLKD1 AND CLKD0

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

5.1.10 RBIT

DESCRIPTION

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

5.1.11 CLKR1 AND CLKR0

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

5.1.12 MCLK1

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

5.1.13 MCLK0

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

5.1.14 MCLKB

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

5.1.15 SCOPE

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

```
435                     TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.
```

5.1.16 HLT

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

5.1.17 TRAPCATCHER

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

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

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

WHERE:

```
CS1,CS2,ER ETC.     = RH11/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

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

```
492                     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. TEST DLT AND TRE BITS

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

6. CLEAR DLT AND TRE

CLEAR BY SETTING TRE IN RSCS1 AND TEST.

7. LOAD RSDB WITH ALL ONES AND ALL ZEROS

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

8. TEST FOR 66 LOCATIONS IN SILO

THIS IS DONE BY PUTTING A BINARY COUNT IN EVERY LOCATION AND

```
549                                     CHECKING THE OUTPUT FOR 66 WORDS.
```

9. TEST DLT ERROR

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

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

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

11. TEST PROGRAM INTERRUPT

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

12. MAINTENANCE TIMING TEST

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

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

13. SECTOR FRACTION TEST

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

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

```
606                                     WHEN THE LAST WORD IS BEING TRANSFERRED, SECTOR AND  FRACTION
607                                     IS  EQUAL  TO  7777  TO  INDICATE  LAST WORD ON THIS TRACK --
```

HANDLE END OF TRACK SPECIAL FOR THE LOOK-AHEAD REGISTER WILL CLEAR THE FRACTION BITS IF ANOTHER WORD IS CLOCKED. RSLA SHOULD INDICATE 7700 ON ANOTHER MAINTENANCE CLOCK.

14. DISK ILLEGAL FUNCTION TEST

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

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

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

16. DRIVE SEARCH TEST 1

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

17. DRIVE SEARCH TEST 2

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

18. REGISTER MODIFICATION REFUSED TEST

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

1. RSCS1
2. RSDA
3. RSER

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

19. DRIVE SELECT TEST

THE PROGRAM LOADS A DRIVE REGISTER, OF THE DRIVE UNDER TEST, TO ALL ONES. THE PROGRAM THEN FINDS A NON-EXISTENT DRIVE AND

```
664                                     TRIES TO LOAD ITS REGISTER WITH  ALTERNATE  ONES  AND  ZEROS.
665                                     THIS  SHOULD  CAUSE  "NED"  TO  SET  IN  RSCS2.   THE PROGRAM
```

RE-SELECTS THE DRIVE UNDER TEST AND CHECKS ITS REGISTER TO SEE IF IT WAS MODIFIED. IT SHOULD CONTAIN ALL ONES.

20. MAINTENANCE WRITE TEST

THIS IS AN RS04 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR WRITE TEST. WE ARE TESTING THE COMPLETE DATA PATH FOR A DATA TRANSFER TO THE DISK. MILLER ENCODED DATA TO BOTH SURFACES IS CHECKED ALONG WITH CORRECT GENERATION OF THE CRC WORD AT THE END OF THE SECTOR. INDEX PULSES, RESYNC, TIMING PREAMBLE, AND SECTOR PULSES ARE ALSO CHECKED.

21. MAINTENANCE READ TEST

THIS IS AN RS04 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR READ TEST. WE ARE TESTING THE COMPLETE DATA PATH FROM THE DISK DECODING LOGIC TO CORE MEMORY. (THE PHASE LOCK LOOP IS NOT TESTED IN MAINTENANCE MODE.)

22. MAINTENANCE MODE DATA WRITE CHECK TEST

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

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

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

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

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

25. IGNORE FUNCTION TEST

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

```
722                                     26.  INVALID ADDRESS TEST
```

FLOAT A 1 THROUGH THE FOUR SPARE ADDRESS BITS IN THE DISK ADDRESS REGISTER (RSDA). THIS SHOULD CAUSE "IAE" TO SET IN THE ERROR REGISTER (RSER) WHEN A READ FUNCTION IS LOADED INTO RSCS1 WHICH IN TURN SHOULD CAUSE ATTENTION TO SET IN THE DRIVE STATUS REGISTER (RSDS) AND "TRE" AND "SC" TO SET IN THE CONTROL REGISTER (RSCS1).

27. DISK OPERATION INCOMPLETE (OPI) ERROR TEST

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

28. PARITY ERROR TEST

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

29. MAINTENANCE MODE INTERRUPT TEST

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

30. DISK ADDRESS OVERFLOW (AOE) TEST

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

31. MAINTENANCE VERIFY TEST

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

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

```
                                      ;        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  021736                $ENDAD
                000052                .=       52
        000052  040000                BIT14

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

                000200                .=       200
        000200  000137  000232                 JMP     @#BEGIN1

                000220                .=       220
        000220  052737  000100  001146         BIS     #BIT6,FLAG3      ;TEST ALL DRIVES
        000226  000137  001240        BEGIN2:  JMP     @#BEGIN

        000232  042737  000100  001146 BEGIN1: BIC     #BIT6,FLAG3      ;CLEAR MULTI DRIVE FLAG
        000240  000772                         BR      BEGIN2
```

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

```
                001000                  .=      1000
                                        ICNT:   0               ;LH = ITERATION COUNT  ;RH = TEST NO.
                                        ERRORS: 0               ;ERROR COUNT
                                        PCNT:   0,0             ;2 WORD PASS COUNT
                                        LAD:    0               ;LOOP ADDRESS FOR SCOPE
                                        HLTADR: 0               ;ADDRESS OF LAST HLT INSTRUCTION  EXECUTED
                                        FILCHR: 1000            ;FILCHR=0 (CHAR)  ;FILCHR+1=2 (COUNT)
                                        TPS:    177564          ;OUTPUT STATUS REGISTER
                                        TKS:    177560
                                        TKB:    177562
                                        TPB:    177566          ;OUTPUT BUFFER
                                        SWR:    177570          ;SWITCH REGISTER
                                        DISPLAY:177570          ;DISPLAY REGISTER

                001100                  .=      1100

                                        ;DISK I/O REGISTERS

                                        RSCS1:  172040          ;DISK CONTROL + STATUS REGISTER
                                        RSCS2:  172050          ;DISK CONTROL + STATUS REGISTER
                                        RSWC:   172042          ;WORD COUNT REGISTER
                                        RSBA:   172044          ;BUS ADDRESS
                                        RSDA:   172046          ;DISK ADDRESS (DESIRED ADDRESS)
                                        RSDS:   172052          ;DRIVE STATUS
                                        RSER:   172054          ;ERROR REG.
                                        RSAS:   172056          ;ATTENTION SUMMARY
                                        RSLA:   172060          ;LOOK AHEAD
                                        RSDB:   172062          ;DATA BUFFER REGISTER
                                        RSMR:   172064          ;MAINTENANCE REGISTER
                                        RSDT:   172066          ;DRIVE TYPE REGISTER
                                        RSVEC:  204             ;INTERUPT VECTOR
                                        RSVCPS: 206             ;INTERRUPT PRIO. VECTOR
                                        RSCS1B: 172041          ;ODD BYTE ADD FOR CS1
                                        RSCS2B: 172051          ;ODD BYTE ADD FOR CS2
                                        RSWCB:  172043          ;ODD BYTE ADD FOR CW
                                        RSBAB:  172045          ;ODD BYTE ADD FOR BA
```

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

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

                        ;BIT ASSIGNMENTS FOR THE REGISTER BITS

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

```
                                ;WORKING LOCATIONS
                                
001144  000000                FLAG2:  0               ;SECOND FLAG WORD
001146  000000                FLAG3:  0               ;3RD FLAG WD
001150  000000                LSTEV:  0               ;LAST EVEN BIT TRANSFERED
001152  000000                LSTOD:  0               ;LAST ODD BIT TRANSFERED
001154  000000                NOWEV:  0               ;PRESENT EVEN BIT BEING XFERED
001156  000000                NOWOD:  0               ;PRESENT ODD BIT BEING XFERED
001160  000000                RS0:    0               ;SAME
001162  000000                UNNUM:  0               ;UNIT CURRENTLY BEING TESTED
001164  000000                UNITSV: 0               ;SET BIT=UNIT ON BUS
001166  000000                UNCMP:  0               ;FOR COMPARING FOR # OF DEVICE
001170  000000                ONCEE:  0               ;DID WE TEST ANY DRIVES
001172  000000                TIMSV:  0               ;SAVE LOC FOR TIME
        172100                MPR0=172100             ;PARITY REG
001174  000000                SAVEE:  0               ;WORK LOC
001176  000000  000000        MCCNT:  0,0             ;MAINT CLOCK COUNT
001202  000000                WCRC:   0               ;WORK LOC FOR CREATING CRC WORD
001204  000000                REPT:   0               ;REPEAT COUNTER
001206  000000                REPT1:  0               ;REPEAT COUNTER
001210  000000                CLKCNT: 0               ;CLOCK COUNTER FOR EACH WORD
001212  000000                INBIT:  0               ;USED IN CRC CAL ROUTINE
001214  000000                WK15:   0               ;USED IN CRC CAL ROUTINE
001216  000000                WORK:   0
001220  000000                WORK0:  0
001222  000000                WORK1:  0
001224  000000                WORK2:  0
001226  000000                WORK3:  0
001230  000000                WORK4:  0
001232  000000                WORK5:  0
001234  000000                WORK6:  0
001236  000000                LAFLAG: 0               ;FLAG FOR LA DONE TYPE OUT
```

```
                        ;DISCRIPTION OF BITS IN LOCATION ONCEE

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

                        ;DISCRIPTION OF BITS IN LOCATION FLAG2

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

```
 988  001240  012706  000500          BEGIN:  MOV     #500,SP         ;SET STACK TO  *** 500 ***
 989  001244  012737  025760  000024          MOV     #.POWER,@#24    ;SET UP PF VECTOR
 990  001252  012737  000340  000026          MOV     #340,@#26       ;LOCK OUT THE WORLD
 991  001260  012737  025406  000030          MOV     #.HLT,@#30      ;SET EMT VECTOR
 992  001266  012737  000340  000032          MOV     #340,@#32       ;LOCK UP
 993  001274  012737  026414  000034          MOV     #.TRAP,@#34     ;SET TRAP VECTOR
 994  001302  012737  000340  000036          MOV     #340,@#36       ;LOCK UP
 995  001310  005037  001000                  CLR     ICNT            ;INIT ICNT
 996  001314  005037  001010                  CLR     LAD             ;INIT LAD
 997  001320  042737  000020  001146          BIC     #BIT4,FLAG3     ;CLEAR TEST ONLY ONE DRIVE FLAG
 998  001326  005037  001236                  CLR     LAFLAG          ;CLEAR TYPE FLAG FOR LA DSK
 999  001332  042737  177677  001144          BIC     #177677,FLAG2
1000  001340  042737  153777  001170          BIC     #153777,ONCEE
1001  001346  032737  000100  001146          BIT     #BIT6,FLAG3     ;TEST ALL DRIVES?
1002  001354  001402                          BEQ     5$              ;ASK
1003  001356  000137  002006                  JMP     @#MULTII
1004  001362                          5$:
1005  001362  104402  001366                  TYPE    ,.+2            ;.ASCIZ <15><12>"TEST ALL DRIVES? (Y OR N) "
1006  001424  104412                          RDLIN
1007  001426  122737  000131  026374          CMPB    #'Y,INPUT       ;TEST FOR YES
1008  001434  001564                          BEQ     MULTII          ;YES
1009  001436  052737  000020  001146          BIS     #BIT4,FLAG3     ;SET TEST ONLY ONE DRIVE FLAG
1010  001444                          1$:
1011  001444  104402  001450                  TYPE    ,.+2            ;.ASCIZ "TYPE UNIT #"
1012  001464  104410                          RDOCT
1013  001466  012604                          MOV     (6)+,R4         ;GET NUMBER
1014  001470  022704  000010                  CMP     #10,R4          ;CORRECT #
1015  001474  101763                          BLOS    1$              ;NO
1016  001476  010437  001162                  MOV     R4,UNNUM        ;SET UNIT #
1017  001502  005002                          CLR     R2              ;CLEAR WORK AREA
1018  001504  000261                          SEC                     ;SET CARRY
1019  001506  006102                  2$:     ROL     R2              ;SET WORK BIT
1020  001510  005704                          TST     R4              ;IS THIS BIT CORRESPOND WITH CORRECT DRIVE #
1021  001512  001402                          BEQ     3$              ;YES
1022  001514  005304                          DEC     R4              ;NO TRY AGAIN
1023  001516  000773                          BR      2$              ;TEST AGAIN
1024  001520  010237  001164          3$:     MOV     R2,UNITSV       ;SET DRIVE BIT IN UNITSV
1025  001524  010237  001166                  MOV     R2,UNCMP        ;SET UNIT COMPARE
1026  001530  013777  001162  177344          MOV     UNNUM,@RSCS2    ;LOAD DRIVE
1027  001536  012777  177777  177350          MOV     #-1,@RSER       ;LOAD ERRORS
1028  001544  022777  000004  177354          CMP     #4,@RSDT        ;RS04LA?
1029  001552  001033                          BNE     6$              ;NO
1030  001554  104402  001560                  TYPE    ,.+2            ;.ASCIZ <15><12>"THIS IS A RS03LA IT WILL BE TESTED AS A
1031  001642                          6$:
1032  001642  104402  001646                  TYPE    ,.+2            ;.ASCIZ "ALL ERROR LIGHTS ON SELECTED UNIT SHOULD BE ON
1033  001756  000000                          HALT                    ;WAIT FOR LIGHTS TO BE CHECKED
1034  001760  023777  001164  177130          CMP     UNITSV,@RSAS    ;DID CORRECT ATA SET
1035  001766  001405                          BEQ     4$
1036  001770  017700  177122                  MOV     @RSAS,BAD       ;GET RSAS
1037  001774  013701  001164                  MOV     UNITSV,GOOD     ;GET CORRECT AND
1038  002000  104000                          HLT                     ;RSAS=BAD GOOD=CORRECTIONS
1039                                                                  ;ATA BIT SHOULD SET FOR ERRORS
1040                                                                  ;WERE SET IN RSER
1041  002002  000137  002564          4$:     JMP     NOWGO           ;START TESTING
```

```
                                        ;NOW TEST FOR DRIVES
                                        
002006  012701  000010          MULTII: MOV     #8.,R1          ;PUT 8 INTO R1 FOR COUNT
002012  005077  177064                  CLR     @RSCS2          ;SET DEVICE TO ZERO
002016  012777  177777  177070  TRY:    MOV     #-1,@RSER       ;CAUSE AN ERROR +SETS BIT IN RSAS REG
002024  005301                          DEC     R1              ;DO A MAXIMUM OF 8 TIMES
002026  001403                          BEQ     DVNUM           ;TESTED FOR ALL DRIVES GET OUT
002030  005277  177046                  INC     @RSCS2          ;INCREMENT DRIVE UNIT
002034  000770                          BR      TRY             ;REPEAT FOR NEXT DRIVE
002036  017737  177054  001164  DVNUM:  MOV     @RSAS,UNITSV    ;SAVE
002044  012737  000401  001166          MOV     #401,UNCMP      ;SETUP TO CMP WITH UNITSV
002052  012737  000000  001162          MOV     #0,UNNUM        ;PUT 0 INTO UNIT NO.
002060  032777  020000  176740          BIT     #BIT13,@SWR     ;INHIBIT TYPE OUT?
002066  001015                          BNE     STTEST          ;YES
002070  104402  002074                  TYPE    ,.+2            ;.ASCIZ <15><12>"TESTING UNIT "
002114  042737  100000  001170          BIC     #BIT15,ONCEE    ;CLEAR ERROR FLAG
002122  033737  001166  001164  STTEST: BIT     UNCMP,UNITSV    ;IS THIS DRIVE ON THE SYSTEM
002130  001510                          BEQ     TRYNX           ;NO
002132  013777  001162  176742          MOV     UNNUM,@RSCS2    ;YES PUT UNIT # INTO CS2
002140  022777  000000  176760  3$:     CMP     #0,@RSDT        ;IS THIS A RS03?
002146  001454                          BEQ     1$              ;YES
002150  022777  000001  176730          CMP     #1,@RSDT        ;IS IT A RS03?
002156  001450                          BEQ     1$              ;YES
002160  022777  000004  176740          CMP     #4,@RSDT        ;RS03LA?
002166  001071                          BNE     TRYNX           ;NO
002170  005737  001236                  TST     LAFLAG          ;1ST TIME
002174  001041                          BNE     1$              ;NO
002176  032777  020000  176622          BIT     #BIT13,@SWR     ;INHIBIT TYPEOUT
002204  001061                          BNE     4$              ;YES
002206  104402  002212                  TYPE    ,.+2            ;.ASCIZ <15><12>
002216  013746  001162                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
002222  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
002224  104402  002230                  TYPE    ,.+2            ;.ASCIZ " IS A RS03LA WILL TEST IT LIKE A RS03"<15><12>
002300  032777  020000  176520  1$:     BIT     #BIT13,@SWR     ;INHIBIT TYPE OUT?
002306  001020                          BNE     4$              ;YES
002310  032737  100000  001170          BIT     #BIT15,ONCEE    ;ANY ERRORS?
002316  001404                          BEQ     5$              ;NO
002320  104402  002324                  TYPE    ,.+2            ;.ASCIZ <15><12><12>
002330                          5$:
002330  013746  001162                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
002334  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
002336  104402  000040                  TYPE    ,40             ;TYPE SPACE
002342  042737  100000  001170          BIC     #BIT15,ONCEE    ;CLEAR ERROR FLAG
002350  000505                  4$:     BR      NOWGO           ;NOW TEST
002352  032737  000020  001146  TRYNX:  BIT     #BIT4,FLAG3     ;MULTI DRIVE
002360  001074                          BNE     DONEE           ;NO
002362  006337  001166          1$:     ASL     UNCMP           ;CHECK NEXT BIT FOR DRIVE
002366  103403                          BCS     CHCKDV          ;DID WE TEST ANY REG?
002370  005237  001162                  INC     UNNUM           ;INC UNIT #
002374  000652                          BR      STTEST          ;CHECK FOR NEXT DRIVE
```

```
1092  002376  032737  000001  001170  CHCKDV: BIT     #BIT0,ONCEE     ;DID WE TEST ANY DRIVES?
1093  002404  001062                          BNE     DONEE           ;YES WE DID TEST A DRIVE
1094  002406  012737  100000  001166          MOV     #100000,UNCMP   ;NO DRIVES TESTED, COULD NOT SET
1095  002414  005037  001162                  CLR     UNNUM           ;ANY AS BITS, THUS DEFAULTS TO
1096  002420  032777  020000  176400          BIT     #BIT13,@SWR     ;INHIBIT TYPE OUT?
1097  002426  001050                          BNE     4$              ;YES
1098  002430  013746  001162                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
1099  002434  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
1100  002436  104402  000040                  TYPE    ,40             ;TYPE SPACE
1101  002442  104402  002446                  TYPE    ,.+2            ;.ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
1102  002540  012737  000001  001166          MOV     #1,UNCMP        ;SETUP TO TEST UNIT 0
1103  002546  000000                          HALT                    ;WAIT
1104  002550  000405                  4$:     BR      NOWGO           ;TEST DRIVE 0
1105  002552  012737  177777  001236  DONEE:  MOV     #-1,LAFLAG      ;SET LA FLAG DRIVE IDENTIFICATION
1106  002560  000137  021672                  JMP     DONE            ;GET OUT
1107
1108                                  ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
1109                                  ;TO CLEAR ALL THE RH AND RS REGISTERS
1110
1111  002564  052737  000001  001170  NOWGO:  BIS     #BIT0,ONCEE     ;SET FOUND DRIVE FLAG
1112  002572  013737  025404  001172          MOV     TIMES,TIMSV     ;SAVE TIME
1113  002600  012737  000001  025404          MOV     #1,TIMES        ;ONLY TEST ONCE
1114                                  ;*****************************************************************************
1115                                  ;TEST 1         RESET TEST FOR REGISTERS
1116                                  ;*****************************************************************************
1117  002606  104400                  TST1:   SCOPE
1118  002610  012737  000340  177776          MOV     #340,@#PS       ;LOCK OUT INTERUPTS
1119  002616  013777  001162  176256          MOV     UNNUM,@RSCS2    ;LOAD UNIT NO.
1120  002624  012777  177776  176246          MOV     #177776,@RSCS1  ;SET ALL
1121  002632  012777  177777  176246          MOV     #177777,@RSBA   ;POSSIBLE R/W
1122  002640  012777  177777  176242          MOV     #177777,@RSDA   ;BITS IN THESE REGISTERS
1123  002646  012777  177777  176240          MOV     #177777,@RSER
1124  002654  012777  177777  176242          MOV     #177777,@RSMR
1125  002662  012777  177777  176214          MOV     #177777,@RSWC
1126  002670  012777  177737  176204          MOV     #177737,@RSCS2
1127  002676  000005                          RESET                   ;CLEAR ALL BITS IN ALL REG.
1128
1129                                  ;TEST RSCS2 FOR CLEARED BITS
1130
1131  002700  022777  000100  176174          CMP     #100,@RSCS2     ;DID THESE BITS GET CLEARED?
1132  002706  001401                          BEQ     .+4             ;YES
1133  002710  104200                          HLT     !CS2            ;(417) SHOULD BE CLEARED IN CS2
1134  002712  013777  001162  176162          MOV     UNNUM,@RSCS2    ;PUT # OF UNIT IN TEST IN CS2
1135  002720  022777  010600  176164          CMP     #10600,@RSDS    ;IS DPR AND MOL SET?
1136  002726  001401                          BEQ     .+4             ;YES
1137  002730  104040                          HLT     !DS             ;NO WHY NOT?
1138
1139                                  ;TEST CONTROL AND STATUS REG 1
1140  002732  022777  004200  176140          CMP     #4200,@RSCS1    ;DID THE READY BIT SET?
1141  002740  001401                          BEQ     .+4             ;YES
1142  002742  104001                          HLT     !CS1            ;READY SHOULD BE SET
```

```
1143                                    ;TEST BUS ADDRESS REGISTER
1144
1145  002744  005777  176136                  TST     @RSBA           ;IS BA REG. CLEARED
1146  002750  001401                          BEQ     .+4             ;YES
1147  002752  104020                          HLT     !BA             ;SHOULD BE 0
1148
1149                                    ;TEST DISK ADDRESS REGISTER
1150
1151  002754  005777  176130                  TST     @RSDA           ;IS DA CLEARED
1152  002760  001401                          BEQ     .+4             ;YES
1153  002762  104004                          HLT     !DA             ;SHOULD BE 0
1154
1155                                    ;TEST ERROR REG RSER
1156
1157  002764  005777  176124                  TST     @RSER           ;DID RSER CLEAR?
1158  002770  001401                          BEQ     .+4             ;YES
1159  002772  104002                          HLT     !ER             ;BITS(157015) SHOULD BE CLEARED
1160
1161                                    ;TEST RS MAINTENANCE REGISTER
1162
1163  002774  032777  000077  176122          BIT     #77,@RSMR       ;DID THESE BITS GET CLEARED
1164  003002  001401                          BEQ     .+4             ;YES
1165  003004  104220                          HLT     !MR             ;BITS(77) SHOULD BE 0
1166
1167                                    ;TEST WC REG IT SHOULD NOT CHANGE
1168
1169  003006  022777  177777  176070          CMP     #177777,@RSWC   ;DID IT CHANGE?
1170  003014  001401                          BEQ     .+4             ;NO
1171  003016  104010                          HLT     !WC             ;RESET SHOULD NOT MODIFY RSWC
1172
1173                                    ;TEST RSAS
1174
1175  003020  005777  176072                  TST     @RSAS           ;IS REG CLEAR
1176  003024  001401                          BEQ     .+4             ;YES
1177  003026  104100                          HLT     !AS             ;NO
```

```
                                ;*****************************************************************
                                ;TEST 2          TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
                                ;*****************************************************************
003030  104400                  TST2:   SCOPE

003032  012737  000340  177776  TTAGG:  MOV     #340,@#PS       ;LOCK OUT INTERRUPTS
003040  013777  001162  176034          MOV     UNNUM,@RSCS2    ;LOAD UNIT NO.
003046  012777  043576  176024          MOV     #43576,@RSCS1   ;SET ALL
003054  012777  020417  176020          MOV     #20417,@RSCS2   ;ALL
003062  012777  177777  176016          MOV     #177777,@RSBA   ;POSSIBLE
003070  012777  177777  176012          MOV     #177777,@RSDA   ;REGISTERS
003076  012777  177017  176010          MOV     #177017,@RSER
003104  012777  177777  176010          MOV     #177777,@RSDB
003112  012777  177777  175764          MOV     #177777,@RSWC
003120  012777  020417  175754          MOV     #20417,@RSCS2
003126  012777  000071  175770          MOV     #71,@RSMR
003134  012777  000040  175740          MOV     #40,@RSCS2      ;CLEAR ALL BITS
003142  022777  000100  175732          CMP     #100,@RSCS2     ;DID THE RIGHT BITS CLEAR?
003150  001401                        BEQ     .+4             ;YES
003152  104200                        HLT     !CS2            ;(417) SHOULD BE CLEARED IN CS2
003154  013777  001162  175720          MOV     UNNUM,@RSCS2    ;GET DRIVE NUMBER
003162  032777  173577  175710          BIT     #173577,@RSCS1  ;DID ALL BITS GET CLEARED
003170  001401                        BEQ     .+4             ;YES
003172  104001                        HLT     !CS1            ;NO, ALL BITS SHOULD BE 0
                                ;TEST BUS ADDRESS REGISTER

003174  005777  175706                TST     @RSBA           ;IS BA REG. CLEARED
003200  001401                        BEQ     .+4             ;YES
003202  104020                        HLT     !BA             ;SHOULD BE 0

                                ;TEST DISK ADDRESS REGISTER

003204  005777  175700                TST     @RSDA           ;IS DA CLEARED
003210  001401                        BEQ     .+4             ;YES
003212  104020                        HLT     !BA             ;SHOULD BE 0

                                ;TEST ERROR REG RSER

003214  032777  177777  175672          BIT     #177777,@RSER   ;DID THESE BITS GET CLEARED
003222  001401                        BEQ     .+4             ;YES
003224  104002                        HLT     !ER             ;BITS(157015) SHOULD BE CLEARED

                                ;TEST RS MAINTENANCE REGISTER
003226  032777  000077  175670          BIT     #77,@RSMR       ;DID THESE BITS GET CLEARED
003234  001401                        BEQ     .+4             ;YES
003236  104220                        HLT     !MR             ;BITS(77) SHOULD BE 0

                                ;TEST WC REG. IT SHOULD NOT CHANGE
003240  022777  177777  175636          CMP     #177777,@RSWC   ;DID WC CHANGE
003246  001401                        BEQ     .+4             ;NO
003250  104010                        HLT     !WC             ;WHY DID IT CHANGE?
```

```
                                ;*****************************************************************************
                                ;TEST 3          SET AND CLEAR ALL REGISTERS
                                ;*****************************************************************************
003252  104400                  TST3:   SCOPE
                                ;CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
                                ;BITS 7,6,5,4,3,2&1

003254  104414                          CLRDK                   ;CLEAR ALL RS REG
003256  013737  001172  025404          MOV     TIMSV,TIMES     ;GET TIME
003264  012777  003576  175606          MOV     #3576,@RSCS1    ;SET DISK FUNCTION BITS
003272  022777  007776  175600          CMP     #7776,@RSCS1    ;ARE THESE BITS SET?
003300  001401                          BEQ     .+4             ;NO
003302  104001                          HLT     !CS1            ;SHOULD = 3776
003304  012777  002524  175566          MOV     #2524,@RSCS1    ;SET THESE BITS
003312  022777  006724  175560          CMP     #6724,@RSCS1    ;DID THEY SET
003320  001401                          BEQ     .+4             ;YES
003322  104001                          HLT     !CS1            ;SHOULD BE 2725
003324  012777  001052  175546          MOV     #1052,@RSCS1    ;SET THESE BITS
003332  022777  005252  175540          CMP     #5252,@RSCS1    ;ARE THEY =?
003340  001401                          BEQ     .+4             ;YES
003342  104001                          HLT     !CS1            ;SHOULD = 1252
003344  104400                  TST4:   SCOPE
                                ;CLEAR THE FUNCTION BITS

003346  012777  043576  175524          MOV     #43576,@RSCS1   ;SET DISK FUNCTION BITS
003354  005077  175520                  CLR     @RSCS1
003360  022777  004200  175512          CMP     #4200,@RSCS1    ;IS THE READY BIT SET
003366  001401                          BEQ     .+4             ;YES
003370  104001                          HLT     !CS1            ;RSCS1 SHOULD = 4200

                                ;*****************************************************************************
                                ;TEST 5          TEST RSCS2
                                ;*****************************************************************************
003372  104400                  TST5:   SCOPE

003374  000005                          RESET                   ;CLEAR WORLD
003376  022777  000100  175476          CMP     #100,@RSCS2     ;DID THEY CLEAR?
003404  001401                          BEQ     .+4             ;YES
003406  104200                          HLT     !CS2            ;NO
003410  012777  021037  175464          MOV     #21037,@RSCS2   ;SET BITS 21017
003416  022777  021137  175456          CMP     #21137,@RSCS2   ;DID THESE BITS GET SET
003424  001405                          BEQ     1$              ;YES
003426  017700  175450                  MOV     @RSCS2,BAD
003432  012701  021137                  MOV     #21137,GOOD     ;WHAT CS2 SHOULD =
003436  104000                          HLT                     ;CS2 = BAD GOOD = CORRECT ANS
```

```
1274  003440  012777  020025  175434  1$:     MOV     #20025,@RSCS2   ;SET THESE BITS
1275  003446  022777  020125  175426          CMP     #20125,@RSCS2   ;DID THESE BITS GET SET
1276  003454  001401                          BEQ     .+4             ;YES
1277  003456  104200                          HLT     !CS2            ;NO,CS2 SHOULD = 20125
1278  003460  012777  000012  175414          MOV     #12,@RSCS2      ;LOAD THESE BITS
1279  003466  022777  000112  175406          CMP     #112,@RSCS2     ;DID THESE BITS GET SET IN CS2
1280  003474  001401                          BEQ     .+4             ;YES
1281  003476  104200                          HLT     !CS2            ;BAD = CS2 GOOD = CORRECT ANS
1282  003500  012777  177777  175374          MOV     #-1,@RSCS2      ;SET BITS
1283  003506  005077  175370                  CLR     @RSCS2          ;CLEAR THEM
1284  003512  022777  000100  175362          CMP     #100,@RSCS2     ;DID CLEAR WORK
1285  003520  001401                          BEQ     .+4             ;YES
1286  003522  104200                          HLT     !CS2            ;R/W BITS DID NOT CLEAR
1287  003524  013777  001162  175350          MOV     UNNUM,@RSCS2    ;GET UNIT #
1288  003532  104400              TST6:       SCOPE
1289                              ;CAN WE SET ALL THE RSBA BITS
1290
1291  003534  012777  177777  175344          MOV     #177777,@RSBA   ;SET THE BITS
1292  003542  022777  177776  175336          CMP     #177776,@RSBA   ;DID THEY SET
1293  003550  001401                          BEQ     .+4             ;YES
1294  003552  104020                          HLT     !BA             ;BITS 17776 SHOULD BE SET
1295  003554  012777  125252  175324          MOV     #125252,@RSBA   ;SET THESE BITS
1296  003562  022777  125252  175316          CMP     #125252,@RSBA   ;ARE THEY =
1297  003570  001401                          BEQ     .+4             ;YES
1298  003572  104020                          HLT     !BA             ;SHOULD BE 125252
1299  003574  012777  052524  175304          MOV     #52524,@RSBA    ;SET THESE BITS
1300  003602  022777  052524  175276          CMP     #52524,@RSBA    ;ARE THEY =
1301  003610  001401                          BEQ     .+4             ;YES
1302  003612  104020                          HLT     !BA             ;SHOULD BE 52524
1303
1304  003614  104400              TST7:       SCOPE
1305                              ;FLOAT A 1 THROUGH RSBA
1306
1307  003616  012701  000002      FLOTBA:     MOV     #2,GOOD         ;GET A 2
1308  003622  000241                          CLC                     ;CLEAR CARRY
1309  003624  010177  175256      1$:         MOV     GOOD,@RSBA      ;FLOAT NUMBER
1310  003630  017700  175252                  MOV     @RSBA,BAD       ;GET BA
1311  003634  020100                          CMP     GOOD,BAD        ;COMPARE BA
1312  003636  001401                          BEQ     .+4             ;BA CORRECT
1313  003640  104000                          HLT                     ;BAD=BA GOOD=CORRECT ANS
1314  003642  006101                          ROL     GOOD            ;ROTATE NUMBER
1315  003644  103367                          BCC     1$              ;LOOP TILL DONE
```

```
1316  003646  104400                  TST10:  SCOPE
1317
1318                                  ;CLEAR THE RSBA REGISTER
1319
1320  003650  012777  177777  175230          MOV     #177777,@RSBA   ;SET RSBA EQUAL TO ALL ONES
1321  003656  005077  175224                  CLR     @RSBA
1322  003662  005777  175220                  TST     @RSBA           ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
1323  003666  001401                          BEQ     .+4             ;YES
1324  003670  104020                          HLT     !BA             ;NO
1325  003672  104400                  TST11:  SCOPE
1326
1327                                  ;CAN WE SET ALL BITS IN RSWC REGISTER
1328
1329  003674  012777  177777  175202          MOV     #177777,@RSWC   ;SET WC BITS
1330  003702  022777  177777  175174          CMP     #177777,@RSWC   ;ARE ALL BITS SET
1331  003710  001401                          BEQ     .+4             ;YES
1332  003712  104010                          HLT     !WC             ;NO
1333  003714  012777  125252  175162          MOV     #125252,@RSWC   ;SET THESE BITS
1334  003722  022777  125252  175154          CMP     #125252,@RSWC   ;ARE THEY =
1335  003730  001401                          BEQ     .+4             ;YES
1336  003732  104010                          HLT     !WC             ;SHOULD BE 125252
1337  003734  012777  052525  175142          MOV     #52525,@RSWC    ;SET THESE BITS
1338  003742  022777  052525  175134          CMP     #52525,@RSWC    ;ARE THEY =
1339  003750  001401                          BEQ     .+4             ;YES
1340  003752  104010                          HLT     !WC             ;SHOULD BE 152525
1341  003754  104400                  TST12:  SCOPE
1342
1343                                  ;FLOAT A 1 THROUGH RSWC
1344
1345  003756  012701  000001          FLOTWC: MOV     #1,GOOD         ;GET A 1
1346  003762  000241                          CLC                     ;CLEAR CARRY
1347  003764  010177  175114          1$:     MOV     GOOD,@RSWC      ;FLOAT NUMBER
1348  003770  017700  175110                  MOV     @RSWC,BAD       ;GET WC
1349  003774  020100                          CMP     GOOD,BAD        ;COMPARE WC
1350  003776  001401                          BEQ     .+4             ;WC CORRECT
1351  004000  104000                          HLT                     ;BAD=WC GOOD=CORRECT ANS
1352  004002  006101                          ROL     GOOD            ;ROTATE NUMBER
1353  004004  103367                          BCC     1$              ;LOOP TILL DONE
```

```
1354
1355  004006  104400                    ;CLEAR THE WORD COUNT REGISTER
                                TST13:  SCOPE
1356
1357  004010  012777  177777  175066          MOV     #177777,@RSWC   ;SET RSWC REGISTER EQUAL TO ALL ONES
1358  004016  005077  175062          CLR     @RSWC
1359  004022  005777  175056          TST     @RSWC           ;DID ALL BITS GET CLEARED
1360  004026  001401                  BEQ     .+4             ;YES
1361  004030  104010                  HLT     !WC             ;NO
1362  004032  104400                  TST14:  SCOPE
1363
1364                                  ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.
1365
1366  004034  012777  177777  175046          MOV     #177777,@RSDA   ;SET ALL BITS
1367  004042  022777  177777  175040          CMP     #177777,@RSDA   ;ARE THE BITS SET
1368  004050  001401                  BEQ     .+4             ;YES
1369  004052  104004                  HLT     !DA             ;NO
1370  004054  012777  125252  175026          MOV     #125252,@RSDA   ;SET THESE BITS
1371  004062  022777  125252  175020          CMP     #125252,@RSDA   ;ARE THEY =
1372  004070  001401                  BEQ     .+4             ;YES
1373  004072  104004                  HLT     !DA             ;SHOULD BE 125252
1374  004074  012777  052525  175006          MOV     #52525,@RSDA    ;SET THESE BITS
1375  004102  022777  052525  175000          CMP     #52525,@RSDA    ;ARE THEY =
1376  004110  001401                  BEQ     .+4             ;YES
1377  004112  104004                  HLT     !DA             ;SHOULD BE 52525
1378  004114  104400                  TST15:  SCOPE
1379
1380                                  ;FLOAT A 1 THROUGH RSDA
1381
1382  004116  012701  000001  FLOTDA: MOV     #1,GOOD         ;GET A 1
1383  004122  000241                  CLC                     ;CLEAR CARRY
1384  004124  010177  174760  1$:     MOV     GOOD,@RSDA      ;FLOAT NUMBER
1385  004130  017700  174754          MOV     @RSDA,BAD       ;GET DA
1386  004134  020100                  CMP     GOOD,BAD        ;COMPARE DA
1387  004136  001401                  BEQ     .+4             ;DA CORRECT
1388  004140  104000                  HLT                     ;BAD=DA GOOD=CORRECT ANS
1389  004142  006101                  ROL     GOOD            ;ROTATE NUMBER
1390  004144  103367                  BCC     1$              ;LOOP TILL DONE
```

```
1391                                        ;CAN WE CLEAR THE RSDA REG.
1392  004146  104400                TST16:  SCOPE
1393
1394  004150  012777  177777  174732        MOV     #177777,@RSDA   ;SET RSDA TO ALL ONES
1395  004156  005077  174726                CLR     @RSDA
1396  004162  005777  174722                TST     @RSDA           ;TEST FOR ZERO RSDA
1397  004166  001401                        BEQ     .+4             ;YES
1398  004170  104004                        HLT     !DA             ;ANS SHOULD BE 0
1399  004172  104400                TST17:  SCOPE
1400
1401                                        ;SET AND CLEAR THE RSER REG.
1402
1403  004174  012777  177017  174712        MOV     #177017,@RSER   ;SET THESE BITS
1404  004202  022777  177017  174704        CMP     #177017,@RSER   ;DID THEY SET
1405  004210  001401                        BEQ     .+4             ;YES
1406  004212  104002                        HLT     !ER             ;RSER SHOULD = 157017
1407  004214  112777  000001  174672        MOVB    #1,@RSER        ;A MOVB INST
1408  004222  022777  000001  174664        CMP     #1,@RSER        ;SHOULD MODIFY COMPLETE WD
1409  004230  001401                        BEQ     .+4             ;OK
1410  004232  104002                        HLT     !ER
1411
1412  004234  104400                TST20:  SCOPE
1413
1414  004236  012777  052005  174650        MOV     #52005,@RSER    ;SET THESE BITS
1415  004244  022777  052005  174642        CMP     #52005,@RSER    ;DID THEY SET
1416  004252  001401                        BEQ     .+4             ;YES
1417  004254  104002                        HLT     !ER             ;ER SHOULD = 52005
1418  004256  104400                TST21:  SCOPE
1419
1420  004260  012777  125012  174626        MOV     #125012,@RSER   ;SET THESE BITS
1421  004266  022777  125012  174620        CMP     #125012,@RSER   ;DID THEY SET
1422  004274  001401                        BEQ     .+4             ;YES
1423  004276  104002                        HLT     !ER             ;ER SHOULD = 105012
```

```
1424  004300  104400                    TST22:  SCOPE
1425
1426  004302  012777  177017  174604            MOV     #177017,@RSER   ;SET THESE BITS
1427  004310  005077  174600                    CLR     @RSER           ;CLEAR THEM
1428  004314  005777  174574                    TST     @RSER           ;DID THEY CLEAR
1429  004320  001401                            BEQ     .+4             ;YES
1430  004322  104002                            HLT     !ER             ;SHOULD = 0
1431  004324  104400                    TST23:  SCOPE
1432
1433                                    ;SET AND CLEAR RSMR
1434
1435  004326  012777  000070  174570            MOV     #70,@RSMR       ;SET THESE BITS
1436  004334  017737  174564  001216            MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
1437  004342  042737  177700  001216            BIC     #177700,WORK    ;CLEAR JUNK
1438  004350  022737  000070  001216            CMP     #70,WORK        ;DID THEY SET
1439  004356  001401                            BEQ     .+4             ;YES
1440  004360  104220                            HLT     !MR             ;SHOULD = 70
1441  004362  104400                    TST24:  SCOPE
1442
1443  004364  012777  000070  174532            MOV     #70,@RSMR       ;SET BITS
1444  004372  005077  174526                    CLR     @RSMR           ;CLEAR THEM
1445  004376  032777  000077  174520            BIT     #77,@RSMR       ;DID THEY CLEAR
1446  004404  001401                            BEQ     .+4             ;YES
1447  004406  104220                            HLT     !MR             ;BITS (77) SHOULD = 0
1448  004410  104400                    TST25:  SCOPE
1449
1450  004412  012777  000050  174504            MOV     #50,@RSMR       ;SET BITS
1451  004420  017737  174500  001216            MOV     @RSMR,WORK      ;PUT IN WORKABLE REG
1452  004426  042737  177700  001216            BIC     #177700,WORK    ;CLEAR JUNK
1453  004434  022737  000050  001216            CMP     #50,WORK        ;DID THESE BITS SET
1454  004442  001401                            BEQ     .+4             ;YES
1455  004444  104220                            HLT     !MR             ;BITS (50) SHOULD BE SET
1456  004446  104400                    TST26:  SCOPE
1457
1458  004450  012777  000020  174446            MOV     #20,@RSMR       ;SET BITS
1459  004456  017737  174442  001216            MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
1460  004464  042737  177700  001216            BIC     #177700,WORK    ;CLEAR JUNK
1461  004472  022737  000020  001216            CMP     #20,WORK        ;DID THEY SET
1462  004500  001401                            BEQ     .+4             ;YES
1463  004502  104220                            HLT     !MR             ;MR SHOULD AT LEAST HAVE A (21)
```

```
                        ;*****************************************************************
                        ;TEST 27                TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
                        ;*****************************************************************
004504  104400          TST27:  SCOPE
004506  104414          BITST:  CLRDK                   ;CLEAR ALL RS REG
004510  012777  003566  174362          MOV     #3566,@RSCS1    ;LOAD CST
004516  112777  000005  174370          MOVB    #5,@RSCS1B      ;LOAD BIT
004524  026777  003566  174346          CMP     #3566,@RSCS1    ;DID IT LOAD?
004532  001401                          BEQ     .+4             ;YES
004534  104001                          HLT     ;CS1
004536  112777  000032  174334          MOVB    #32,@RSCS1
004544  026777  006632  174326          CMP     #6632,@RSCS1
004552  001401                          BEQ     .+4
004554  104001                          HLT     ;CS1            ;CS1 SHOULD = 6632
004556  104400          TST30:  SCOPE
004560  013777  001162  174314  BITCS2: MOV     UNNUM,@RSCS2    ;LOAD UNIT NUMBER
004566  012777  177400  174306          MOV     #177400,@RSCS2  ;LOAD ALL BYTE
004574  105077  174336                  CLRB    @RSCS2          ;CLR UPPER BYTE
004600  013701  001162                  MOV     UNNUM,R0        ;GET UNIT NO.
004604  050001                          BIS     R0,R1           ;GET OR BIT
004606  012700  174266                  MOV     #RSCS2,R0       ;GET CS2
004612  020001                          CMP     R0,R1           ;IS CS2 CORRECT?
004614  001401                          BEQ     .+4             ;YES
004616  104000                          HLT                     ;LOAD BYTE DID NOT WORK
004620  104400          TST31:  SCOPE
004622  012777  005252  174252  BITWC:  MOV     #5252,@RSWC     ;LOAD WC
004630  112777  000377  174300          MOVB    #377,@RSCS1     ;LOAD BIT
004636  026777  177652  174236          CMP     #177652,@RSWC   ;DID IT LOAD?
004644  001401                          BEQ     .+4             ;YES
004646  104010                          HLT     ;WC             ;NO WC SHOULD =177652
004650  112777  000123  174224          MOVB    #123,@RSWC
004656  026777  177523  174216          CMP     #177523,@RSWC
004664  001401                          BEQ     .+4
004666  104010                          HLT     ;WC             ;WC SHOULD = 177523
004670  104400          TST32:  SCOPE
004672  012777  005252  174204  BITBA:  MOV     #5252,@RSBA     ;LOAD BA
004700  112777  000377  174212          MOVB    #377,@RSCS1     ;LOAD BIT
004706  026777  177652  174170          CMP     #177652,@RSBA   ;DID IT LOAD?
004714  001401                          BEQ     .+4             ;YES
004716  104020                          HLT     ;BA             ;BA SHOULD =177652
004720  112777  000125  174156          MOVB    #125,@RSBA
004726  026777  177624  174150          CMP     #177624,@RSBA
004734  001401                          BEQ     .+4
004736  104020                          HLT     ;BA             ;BA SHOULD = 177525
004740  104414                          CLRDK                   ;CLEAR ALL RS REG
004742
```

```
1516
1517                                    ;*****************************************************************************
1518                                    ;TEST 33                    TEST DATA LATE IN CS2
1519                                    ;*****************************************************************************
1520    004744  104400                  TST33:  SCOPE
1521
1522                                    ;DO A READ FROM SILO: SHOULD GET DLT + TRE ERROR BECAUSE SILO IS EMPTY
1523
1524    004746  104414                  SILOB:  CLRDK                           ;CLEAR ALL RS REG
1525    004750  017700  174146                  MOV     @RSDB,BAD               ;READ FROM EMPTY SILO
1526    004754  017700  174122                  MOV     @RSCS2,BAD              ;GET CS2
1527    004760  012701  100100                  MOV     #100100,GOOD            ;GET CORRECT ANS
1528    004764  053701  001162                  BIS     UNNUM,GOOD              ;FOR CS2
1529    004770  020001                          CMP     BAD,GOOD                ;IS CS2 CORRECT?
1530    004772  001401                          BEQ     .+4                     ;YES
1531    004774  104200                          HLT     !CS2                    ;SHOULD HAVE DLT ERROR
1532    004776  022777  144200  174074          CMP     #144200,@RSCS1          ;DID SC AND TRE SET?
1533    005004  001401                          BEQ     .+4                     ;YES
1534    005006  104001                          HLT     !CS1                    ;SC AND TRE SHOULD BE SET
1535    005010  012777  040000  174062          MOV     #TRE,@RSCS1             ;CLEARS ERROR BIT
1536    005016  032777  140000  174054          BIT     #140000,@RSCS1          ;DID SC + TRE CLEAR
1537    005024  001401                          BEQ     .+4                     ;YES
1538    005026  104001                          HLT     !CS1                    ;TRE AND SC SHOULD BE 0
1539    005030  017700  174046                  MOV     @RSCS2,BAD              ;GET CS2
1540    005034  042701  100000                  BIC     #BIT15,GOOD             ;GET CORRECT ANS
1541    005040  020100                          CMP     GOOD,BAD                ;IS CS2 CORRECT?
1542    005042  001401                          BEQ     .+4                     ;YES
1543    005044  104200                          HLT     !CS2                    ;DLT SHOULD BE 0
1544                                    ;*****************************************************************************
1545                                    ;TEST 34                    LOAD RSDB WITH ALL ONES AND ALL ZEROS
1546                                    ;*****************************************************************************
1547    005046  104400                  TST34:  SCOPE
1548
1549    005050  104414                  ZERONE: CLRDK                           ;CLEAR ALL RS REG
1550    005052  005077  174044                  CLR     @RSDB                   ;LOAD DB WITH ALL 0
1551    005056  012777  177777  174036          MOV     #177777,@RSDB           ;LOAD DB WITH ALL ONES
1552    005064  012737  002000  001216          MOV     #2000,WORK              ;TIME OUT ROUTINE
1553    005072  012701  000300                  MOV     #300,GOOD               ;GET CORRECT FOR CS2
1554    005076  053701  001162                  BIS     UNNUM,GOOD
1555    005102  017700  173774          2$:     MOV     @RSCS2,BAD              ;GET CS2
1556    005106  020100                          CMP     GOOD,BAD                ;IS IT CORRECT?
1557    005110  001404                          BEQ     3$                      ;YES
1558    005112  005337  001216                  DEC     WORK                    ;TO WAIT FOR OR
1559    005116  001371                          BNE     2$                      ;TO SET
1560    005120  104200                          HLT     !CS2                    ;OR SHOULD BE SET
1561    005122  005001                  3$:     CLR     GOOD
1562    005124  017700  173772                  MOV     @RSDB,BAD               ;LOAD BAD WITH DB
1563    005130  020100                          CMP     GOOD,BAD                ;IS BAD CORRECT
1564    005132  001401                          BEQ     .+4                     ;YES
1565    005134  104000                          HLT                             ;COULD NOT FLOAT 0 THROUGH DB
1566    005136  012701  177777                  MOV     #-1,GOOD                ;LOAD GOOD WITH ANS
1567    005142  017700  173754                  MOV     @RSDB,BAD               ;GET DATA FROM DB
1568    005146  020100                          CMP     GOOD,BAD                ;IS DB CORRECT
1569    005150  001401                          BEQ     .+4                     ;YES
1570    005152  104000                          HLT                             ;BAD SHOULD = 1777777
```

```
1570  005154  104400                    TST35:  SCOPE
1571                                    ;TEST FOR 66 LOCATIONS IN SILO PUT COUNT IN EVERY LOCATION
1572
1573  005156  104414                    SILO:   CLRDK                   ;CLEAR ALL RS REG
1574  005160  005001                            CLR     R1              ;CLEAR COUNTER
1575  005162  005201                    1$:     INC     R1              ;INCREMENT COUNTER
1576  005164  010177  173732                    MOV     R1,@RSDB        ;LOAD SILO
1577  005170  022701  000102                    CMP     #66.,R1         ;LAST LOC. YET?
1578  005174  001372                            BNE     1$              ;NO LOOP AGAIN
1579  005176  012701  000200                    MOV     #200,GOOD       ;GET CORRECT ANS FOR CS2
1580  005202  053701  001162                    BIS     UNNUM,GOOD
1581  005206  017700  173670                    MOV     @RSCS2,BAD      ;GET CS2
1582  005212  020100                            CMP     GOOD,BAD        ;IS CS2 CORRECT?
1583  005214  001401                            BEQ     .+4             ;YES
1584  005216  104200                            HLT     !CS2            ;OR SHOULD BE 1
1585  005220  005001                            CLR     GOOD            ;CLEAR LOCATION COUNTER
1586  005222  005201                    2$:     INC     GOOD            ;ADD 1 TO IT
1587  005224  022701  000103                    CMP     #67.,GOOD       ;LAST LOC YET?
1588  005230  001405                            BEQ     3$              ;YES
1589  005232  017700  173664                    MOV     @RSDB,BAD       ;GET LOC FROM DB
1590  005236  020100                            CMP     GOOD,BAD        ;DO LOCATIONS MATCH?
1591  005240  001770                            BEQ     2$              ;YES
1592  005242  104000                            HLT                     ;CAN NOT MATCH 66 LOCATIONS
1593  005244  032777  000200  173630    3$:     BIT     #OR,@RSCS2      ;IS OR 0
1594  005252  001401                            BEQ     .+4             ;YES
1595  005254  104200                            HLT     !CS2            ;OR SHOULD BE 0
1596
1597                                    ;NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT ERROR
1598
1599  005256  005001                            CLR     R1              ;CLEAR COUNTER
1600  005260  005201                    4$:     INC     R1              ;ADD 1 TO COUNT
1601  005262  010177  173634                    MOV     R1,@RSDB        ;PUT INTO COUNTER
1602  005266  022701  000103                    CMP     #67.,R1         ;DONE YET?
1603  005272  001401                            BEQ     .+4             ;YES
1604  005274  000771                            BR      4$              ;NO DO AGAIN
1605  005276  032777  100000  173576          BIT     #DLT,@RSCS2     ;DID DATA LATE SET?
1606  005304  001001                            BNE     .+4             ;YES
1607  005306  104200                            HLT     !CS2            ;DLT DID NOT SET
1608
1609                                    ;DOES SILO CHANGE WITH 67TH WORD: IT SHOULD NOT
1610
1611  005310  017700  173606                    MOV     @RSDB,BAD       ;GET 1ST WD FORM SILO
1612  005314  012701  000001                    MOV     #1,GOOD         ;CORRECT ANS OF SILO
1613  005320  020100                            CMP     GOOD,BAD        ;IS SILO GOOD
1614  005322  001401                            BEQ     .+4             ;YES
1615  005324  104000                            HLT                     ;SILO SHOULD NOT HAVE MOVED
1616  005326  104400                    TST36:  SCOPE
```

```
1617                                    ;FLOAT A 1 AND A 0 THROUGH THE SILO
1618
1619  005330  104414                    SILOFL: CLRDK                   ;CLEAR ALL RS REG
1620  005332  000241                            CLC                     ;CLEAR CARRY TO FLOAT A 0
1621  005334  012701  000001                    MOV     #1,GOOD         ;GET UP DATA FOR INPUT TO SILO
1622  005340  010177  173556            1$:     MOV     GOOD,@RSDB      ;LOAD DB
1623  005344  006101                            ROL     GOOD            ;SHIFT BIT
1624  005346  103401                            BCS     .+4             ;DONE YET SHIFTING?
1625  005350  000773                            BR      1$              ;NO
1626  005352  012701  177776                    MOV     #-2,GOOD        ;SET ALL ONES
1627  005356  000261                            SEC                     ;SET CARRY TO ROL
1628  005360  010177  173536            3$:     MOV     GOOD,@RSDB      ;LOAD SILO
1629  005364  006101                            ROL     GOOD            ;SHILT 0
1630  005366  103774                            BCS     3$              ;LOOP TILL DONE
1631
1632                                    ;NOW TEST OUTPUT
1633  005370  000241                            CLC                     ;CLEAR CARRY
1634  005372  012701  000001                    MOV     #1,GOOD         ;CORRECT ANS
1635  005376  017700  173520            2$:     MOV     @RSDB,BAD       ;GET DATA FROM DB
1636  005402  020100                            CMP     GOOD,BAD        ;IS DB DATA GOOD?
1637  005404  001401                            BEQ     .+4             ;YES
1638  005406  104000                            HLT                     ;DB COULD NOT BUBBLE CORRECTLY
1639  005410  006101                            ROL     GOOD            ;SETUP FOR NEXT ANS
1640  005412  103401                            BCS     .+4             ;DONE YET?
1641  005414  000770                            BR      2$              ;NO
1642  005416  012701  177776                    MOV     #-2,GOOD        ;SETUP FOR ANS
1643  005422  017700  173474            4$:     MOV     @RSDB,BAD       ;GET DATA FROM DB
1644  005426  020100                            CMP     GOOD,BAD        ;IS IT CORRECT?
1645  005430  001401                            BEQ     .+4             ;YES
1646  005432  104000                            HLT                     ;DB WRONG
1647  005434  000261                            SEC                     ;SET CARRY TO ROL
1648  005436  006101                            ROL     GOOD            ;SETUP FOR NEXT ANS
1649  005440  103770                            BCS     4$              ;LOOP TILL DONE
1650                                    ;TEST INTERRUPT IN THE RH11
1651                                    ;BY MOVING 300 INTO RHCS1
1652                                    ;*****************************************************************************
1653                                    ;TEST 37                TEST INTERRUPT IN RH11
1654                                    ;*****************************************************************************
1655  005442  104400                    TST37:  SCOPE
1656  005444  104414                    INT:    CLRDK                   ;CLEAR ALL ERRORS
1657  005446  012777  005520  173454            MOV     #PGTRAP,@RSVEC  ;SET UP VECTOR
1658  005454  012777  000340  173450            MOV     #340,@RSVCPS    ;SET TRAP PS
1659  005462  012737  000200  177776            MOV     #200,@#PS       ;SET PS AT PRIORITY 4
1660  005470  012777  000300  173402            MOV     #300,@RSCS1     ;THIS SHOULD CAUSE A TRAP
1661  005476  012737  000500  001216            MOV     #500,WORK       ;SETUP LOOP
1662  005504  005337  001216            1$:     DEC     WORK            ;DEC LOOP SHOULD
1663  005510  001375                            BNE     1$              ;INTERRUPT BEFORE LOOP IS DONE
1664  005512  104001                            HLT     !CS1            ;SHOULD NEVER GET HERE
1665  005514  000137  005534                    JMP     INTDON          ;GET OUT
1666  005520  022626                    PGTRAP: CMP     (6)+,(6)+       ;TRAP OK
1667  005522  022777  004200  173350            CMP     #4200,@RSCS1    ;DID IE CLEAR?
1668  005530  001401                            BEQ     .+4             ;YES
1669  005532  104001                            HLT     !CS1            ;IE SHOULD BE CLEARED
1670  005534                            INTDON:
```

```
                                ;*****************************************************************************
                                ;TEST 40                    MAINTENANCE TIMING TEST
                                ;*****************************************************************************
005534  104400                  TST40:  SCOPE

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

                                ;PUT DRIVE IN MAINTENANCE MODE
005536  104414                  MRTIME: CLRDK                   ;CLEAR DRIVE REGISTERS
005540  052737  001040  001170          BIS     #1040,ONCEE     ;SET CLK CNT
005546  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
005550  104420                          MRCK                    ;CHECK MAINTENANCE REG FOR
005552  022701                          22701                   ;22701
005554  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
005556  104430                          MRIND                   ;SEND MAINT INDEX PULSE

005560  104420                          MRCK                    ;CHECK MAINT REG TO
005562  022701                          22701                   ;EQUAL 22701
005564  104000                          HLT                     ;MR=BAD GOOD=CORRECTIONS
                                                                ;COULD NOT INITIALIZE MR REG

                                ;INDEX PULSE SHOULD CLEAR LOOK-AHEAD REG

005566  005777  173326                  TST     @RSLA           ;IS RSLA CLEARED
005572  001401                          BEQ     .+4             ;YES
005574  104224                          HLT     !MR!LA          ;RSLA SHOULD BE CLEARED
                                                                ;WITH THE INDEX PULSE

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

005576  012737  001000  001204          MOV     #512.,REPT
005604  104446                  MRTIM1: MCLK1                   ;CLOCK MAINT REG WITH AN 11
005606  104420                          MRCK                    ;CHECK MR REG TO
005610  032711                          32711                   ;EQUAL 32711
005612  104000                          HLT                     ;MR = BAD GOOD = CORRECT ANS
005614  104450                          MCLK0                   ;CLOCK MR WITH A 1
005616  104420                          MRCK                    ;CHECK MR TO
005620  022701                          22701                   ;EQUAL 22701
005622  104000                          HLT                     ;BAD=MR REG GOOD=CORRECTIONS
005624  005337  001204                  DEC     REPT            ;IS THE LOOP DONE YET?
005630  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

005632  104446                                MCLK1                   ;CLOCK MAINT REG WITH AN 11
005634  104420                                MRCK                    ;CHECK MR REG TO
005636  032311                                32311                   ;EQUAL 32311
005640  104000                                HLT                     ;MR=BAD GOOD=CORRECTIONS
005642  104450                                MCLK0                   ;CLOCK MR WITH A 1
005644  104420                                MRCK                    ;CHECK MAINT REG
005646  022301                                22301                   ;TO EQUAL 22301
005650  104000                                HLT                     ;MR=BAD GOOD-CORRECT ANS
005652  005777  173242                        TST     @RSLA           ;DOES LOOK AHEAD REG=0
005656  001401                                BEQ     MRT2            ;YES-CONT
005660  104224                                HLT     !MR!LA          ;LOOK AHEAD REG SHOULD=0
                                        ;PERFORM MAINTENANCE CLOCK OPERATION 40 TIMES TO PROVIDE
                                        ;CLOCK PULSES TO STEP THRU 1ST SECTOR PRE-AMBLE AREA

005662  005002                        MRT2:   CLR     R2              ;CLEAR R2 FOR SECTOR COMPARE WITH LA REG
005664  012737  000050  001204                MOV     #40.,REPT       ;40 CLOCKS TO STEP THRU PRE-AMBLE
005672  104446                        MRT2A:  MCLK1                   ;CLOCK MR WITH AN 11
005674  104420                                MRCK                    ;CHECK MAINT REG
005676  033711                                33711                   ;EQUAL 33711
005700  104000                                HLT                     ;MR = BAD GOOD = CORRECT ANS
005702  104450                                MCLK0                   ;CLOCK MR REG WITH A 1
005704  104420                                MRCK                    ;CHECK MR REG
005706  023701                                23701                   ;TO EQUAL 23701
005710  104000                                HLT                     ;MR = BAD GOOD = CORRECTANS
005712  005337  001204                        DEC REPT                ;REPEAT
005716  001365                                BNE     MRT2A           ;LOOP 40 TIMES

                                        ;SUPPLY CLOCKS TO STEP THROUGH THE DATA AREA IN THE SECTOR
005720  012737  002200  001204                MOV     #18.*64.,REPT   ;18 CLOCKS PER DATA WORD
005726  104446                        MRT2B:  MCLK1                   ;CLOCK MR WITH AN 11
005730  104420                                MRCK                    ;CHECK MAINT REG
005732  033711                                33711                   ;TO EQUAL 33711
005734  104000                                HLT                     ;MR = BAD GOOD = CORRECT ANS
005736  104450                                MCLK0                   ;CLOCK MR REG WITH A 1
005740  104420                                MRCK                    ;CHECK MR REG
005742  023701                                23701                   ;TO EQUAL 23701
005744  104000                                HLT                     ;MR=BAD GOOD=CORRECTANS
005746  005337  001204                        DEC     REPT            ;REPEAT
005752  001365                                BNE     MRT2B           ;LOOP
```

```
1763                                    ;SUPPLY ENOUGH MAINT CLOCKS TO STEP THROUGH THE CRC AREA
1764                                    ;AND THE DEAD BAND ON THE SECTOR
1765
1766  005754  012737  000214  001204            MOV     #140.,REPT      ;AMOUNT OF CLOCKS TO END OF SECTOR
1767  005762  104446                    MRT2C:  MCLK1                   ;CLOCK MR WITH AN 11
1768  005764  104420                            MRCK                    ;CHECK MAINT REG
1769  005766  033711                            33711                   ;TO EQUAL 33711
1770  005770  104000                            HLT                     ;MR = BAD GOOD = CORRECT ANS
1771  005772  104450                            MCLK0                   ;CLOCK MR REG WITH A 1
1772  005774  104420                            MRCK                    ;CHECK MAINT REG
1773  005776  023701                            23701                   ;TO EQUAL 23701
1774  006000  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
1775  006002  005337  001204                   DEC     REPT            ;REPEAT
1776  006006  001365                            BNE     MRT2C           ;LOOP
1777  006010  104446                            MCLK1                   ;CLOCK MR REG WITH 11
1778  006012  104420                            MRCK                    ;CHECK MR REG
1779  006014  033711                            33711                   ;TO EQUAL 33711
1780  006016  104000                            HLT                     ;MR = BAD GOOD = CORRECT ANS
1781                                    ;ONE MORE CLOCK SHOULD CAUSE SECTOR PULSE
1782                                    ;IF NOT, CHECK E16-6
1783
1784  006020  104450                            MCLK0                   ;CLOCK MR WITH A 1
1785  006022  104420                            MRCK                    ;MAINT REG SHOULD
1786  006024  023701                            23701                   ;EQUAL 23701
1787  006026  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
1788  006030  104446                            MCLK1                   ;CLOCK MR WITH AN 11
1789  006032  104420                            MRCK                    ;CHECK MAINT REG
1790  006034  032311                            32311                   ;SHOULD EQUAL 32311
1791  006036  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
1792
1793                                    ;LOOK-AHEAD REGISTER SHOULD NOW POINT TO SECTOR 1 (OR 4000 IF INTERLEAVED)
1794
1795  006040  022777  000000  173060            CMP     #0,@RSDT        ;INTERLEAVED?
1796  006046  001403                            BEQ     3$              ;NO
1797  006050  062702  004000                   ADD     #4000,R2        ;YES
1798  006054  000402                            BR      2$              ;CONT
1799  006056  062702  000100            3$:     ADD     #100,R2         ;INCREMENT SECTOR COMPARE
1800  006062  020277  173032            2$:     CMP     R2,@RSLA        ;LA REG SHOULD=100
1801  006066  001401                            BEQ     1$              ;LA IS CORRECT
1802  006070  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.

006072  012737  000076  001206  1$:     MOV     #62.,REPT1
006100  012737  002465  001204  MRT3:   MOV     #1333.,REPT
006106  104452                  3$:     MCLKB                   ;CLOCK MR WITH A 1 AND A 11
006110  005337  001204                  DEC     REPT            ;STEP THROUGH
006114  001374                          BNE     3$              ;SECTOR
006116  104450                          MCLKO                   ;CLOCK MR WITH A 1
006120  104420                          MRCK                    ;MAINT REG
006122  022701                          22701                   ;SHOULD EQUAL 22701
006124  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
006126  104446                          MCLK1                   ;1 MORE CLK ASSERTS SECTOR PULSE
006130  104420                          MRCK                    ;MAINT REG SHOULD
006132  032311                          32311                   ;EQUAL 32311
006134  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
006136  022777  000000  172762          CMP     #0,@RSDT        ;DRIVE INTERLEAVED?
006144  001420                          BEQ     6$              ;YES
006146  032737  001000  001170          BIT     #BIT9,ONCEE     ;DO I SET 4000
006154  001406                          BEQ     4$              ;OR CLEAR IT IN RSLA
006156  042737  001000  001170          BIC     #BIT9,ONCEE
006164  162702  004000                  SUB     #4000,R2
006170  000406                          BR      6$
006172  052737  001000  001170  4$:     BIS     #BIT9,ONCEE
006200  062702  004000                  ADD     #4000,R2
006204  000402                          BR      5$
006206  062702  000100          6$:     ADD     #100,R2         ;INCREMENT SECTOR COMPARE
006212  017700  172702          5$:     MOV     @RSLA,BAD       ;LA REG SHOULD HAVE INCREMENTED TO NEXT SECTOR
006216  010201                          MOV     R2,GOOD         ;GET CORRECT ANS FOR RSLA
006220  020100                          CMP     GOOD,BAD        ;COMPARE FOR CORRECT ANS
006222  001401                          BEQ     1$              ;RSLA IS GOOD
006224  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS

006226  005337  001206          1$:     DEC     REPT1           ;REPEAT 62
006232  001322                          BNE     MRT3            ;TIMES
006234  012737  002465  001204          MOV     #1333.,REPT     ;COUNT FOR LAST SECTOR
006242  104452                  2$:     MCLKB                   ;CLOCK
006244  005337  001204                  DEC     REPT            ;THRU
006250  001374                          BNE     2$              ;LAST SECTOR
006252  017700  172642                  MOV     @RSLA,BAD       ;GET CONTENTS OF RSLA
006256  012701  007777                  MOV     #7777,GOOD      ;GET CORRECT ANS
006262  020100                          CMP     GOOD,BAD        ;DOES RSLA EQUAL 7777
006264  001401                          BEQ     .+4             ;YES
006266  104000                          HLT                     ;BAD=RSLA GOOD=CORRECT ANS
```

```
                                ;*****************************************************************************
                                ;TEST #1                    SECTOR FRACTION TEST
                                ;*****************************************************************************
006270  104400                  TST41:  SCOPE

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

006272  104414                  MRT4:   CLRDK                   ;CLEAR DRIVE REGISTERS
006274  052737  000040  001170          BIS     #40,ONCEE       ;SET FLAG BITS
006302  042737  003000  001170          BIC     #3000,ONCEE
006310  005037  001176                  CLR     MCCNT           ;CLEAR MAINT CLOCK COUNTER
006314  005002                          CLR     R2              ;CLEAR R2 FOR SECTOR COUNTER
006316  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
006320  104420                          MRCK                    ;CHECK MAINTENANCE REG FOR
006322  022701                          22701                   ;22701
006324  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
006326  104430                          MRIND                   ;ISSUE A MAINT INDEX PULSE
                                                                ;TO CLEAR THE DRIVE
006330  104420                          MRCK                    ;CHECK MAINT REG
006332  022701                          22701                   ;TO EQUAL 22701
006334  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS

                                ;ISSUE 512 MAINT CLOCKS TO STEP THROUGH THE RESYNC AREA

006336  012737  001000  001204          MOV     #512.,REPT      ;COUNT TO STEP THRU RESYNC AREA
006344  104446                  MRT4A:  MCLK1                   ;CLOCK THROUGH RESYNC
006346  104420                          MRCK                    ;CHECK MAINT REG
006350  032711                          32711                   ;TO EQUAL 32711
006352  104000                          HLT                     ;MR = BAD GOOD = CORRECT ANS
006354  104450                          MCLK0                   ;CLOCK MR REG
006356  104420                          MRCK                    ;CHECK MR REG
006360  022701                          22701                   ;TO EQUAL 22701
006362  104000                          HLT                     ;BAD=MR GOOD=CORRECT ANS
006364  022777  000000  172526          CMP     #0,@RSLA        ;LOOK AHEAD REG
006372  001401                          BEQ     .+4             ;EQUAL 0
006374  104204                          HLT     !LA
006376  005337  001204                  DEC     REPT            ;LOOP THROUGH
006402  001360                          BNE     MRT4A           ;RESYNC AREA

                                ;ONE MORE PULSE SHOULD CAUSE THE FIRST SECTOR PULSE

006404  104446                          MCLK1                   ;CLOCK MR WITH AN 11
006406  104420                          MRCK                    ;CHECK MAINT REG FOR SECTOR PULSE
006410  032311                          32311                   ;MR SHOULD=32311
006412  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
1902  006414  104450                  MRT4B:  MCLKD                   ;CLOCK MR REG WITH A 1
1903  006416  104420                          MRCK                    ;CHECK MAINT REG
1904  006420  022301                          22301                   ;TO EQUAL 22301
1905  006422  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
1906
1907                                  ;SECTOR FRACTION BITS IN LOOK-AHEAD REGISTER SHOULD BE CLEARED (EQUAL TO 00)
1908
1909  006424  017700  172470                  MOV     @RSLA,BAD       ;GET RSLA
1910  006430  010201                          MOV     R2,GOOD         ;GET CORRECT ANS
1911  006432  020100                          CMP     GOOD,BAD        ;IS THE RSLA REG CORRECT
1912  006434  001401                          BEQ     1$              ;YES
1913  006436  104000                          HLT                     ;RSLA=BAD GOOD=CORRECTANS
1914
1915                                  ;STEP THROUGH THE PREAMBLE AREA AND SECTOR DATA
1916                                  ;AREA WHILE CHECKING THE SECTOR FRACTION
1917
1918  006440  012737  000122  001204  1$:     MOV     #82.,REPT       ;FOR FIRST FRACTION CHANGE
1919  006446  104422                  MRT4C:  MRCLK                   ;CLOCK MR REG WITH AN 11 AND A 1
1920  006450  017700  172444                  MOV     @RSLA,BAD       ;GET RSLA
1921  006454  010201                          MOV     R2,GOOD         ;GET CORRECT ANS
1922  006456  020001                          CMP     BAD,GOOD        ;IS RSLA CORRECT
1923  006460  001401                          BEQ     1$              ;YES
1924  006462  104000                          HLT                     ;BAD=RSLA GOOD=CORRECT ANS
1925  006464  005337  001204          1$:     DEC     REPT            ;LOOP ON
1926  006470  001366                          BNE     MRT4C           ;PREAMBLE AREA
1927
1928                                  ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE
1929
1930  006472  104422                          MRCLK                   ;CLOCK MR WITH AN 11 AND A 1
1931  006474  005202                          INC     R2              ;COUNT THE FRACTION
1932  006476  017700  172416                  MOV     @RSLA,BAD       ;GET RSLA
1933  006502  010201                          MOV     R2,GOOD         ;GET CORRECT ANS
1934  006504  020001                          CMP     BAD,GOOD        ;IS RSLA CORRECT?
1935  006506  001401                          BEQ     2$              ;YES
1936  006510  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS
1937
1938                                  ;FIRST FRACTION CHANGES AFTER 82 MAINT CLKS, THE REST
1939                                  ;CHANGE AFTER 20 MAINTENANCE CLOCKS
1940
1941  006512  012737  000076  001204  2$:     MOV     #62.,REPT       ;COUNT FOR WORDS IN A SECTOR
1942  006520  012737  000023  001206  MRT4D:  MOV     #19.,REPT1      ;COUNT FOR SECT FRACT TO CHANGE
1943  006526  104422                  MRT4E:  MRCLK                   ;CLOCK MR WITH AN 11 AND A 1
1944  006530  017700  172364                  MOV     @RSLA,BAD       ;GET RSLA
1945  006534  010201                          MOV     R2,GOOD         ;GET CORRECT ANS
1946  006536  020100                          CMP     GOOD,BAD        ;IS RSLA CORRECT?
1947  006540  001401                          BEQ     1$              ;YES
1948  006542  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS
1949  006544  005337  001206          1$:     DEC     REPT1           ;LOOP
1950  006550  001366                          BNE     MRT4E
```

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

006552  104422                          MRCLK                   ;CLOCK MR WITH AN 11 AND A 1
006554  022702  007777                  CMP     #7777,R2        ;AT THE LAST SECTOR-LAST FRACTION?
006560  001472                          BEQ     MRT4F           ;YES, FINISH THE SECTOR
006562  005202                          INC     R2              ;NO, ADD 1 TO FRACTION
006564  017700  172330          4$:     MOV     @RSLA,BAD       ;GET RSLA
006570  022777  000000  172330          CMP     #0,@RSDT        ;IS THIS DRIVE INTERLEAVED?
006576  001431                          BEQ     12$             ;NO
006600  032737  002000  001170          BIT     #BIT10,ONCEE    ;HAS REPT GONE TO ZERO YET FOR THIS SECTOR?
006606  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).
006610  032737  001000  001170          BIT     #BIT9,ONCEE     ;SHOULD RSLA BE BETWEEN 0000-3700?
006616  001004                          BNE     9$              ;YES
006620  052737  001000  001170          BIS     #BIT9,ONCEE     ;SET FOR NEXT PASS
006626  000406                          BR      10$             ;
006630  042737  001000  001170  9$:     BIC     #BIT9,ONCEE     ;CLEAR FOR NEXT PASS
006636  042702  004000                  BIC     #4000,R2        ;MAKE RSLA LESS THAN 4000
006642  000404                          BR      5$
006644  062702  004000          10$:    ADD     #4000,R2        ;COMPENSATE FOR
006650  162702  000100                  SUB     #100,R2         ;INTERLEAVING
006654  042737  002000  001170  5$:     BIC     #BIT10,ONCEE    ;CLEAR FLAG FOR NEXT SECTOR
006662  010201                  12$:    MOV     R2,GOOD         ;GET CORRECT ANSWER FOR RSLA
006664  020100                          CMP     GOOD,BAD        ;IS RSLA CORRECT
006666  001401                          BEQ     2$              ;YES
006670  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS
006672  005337  001204          2$:     DEC     REPT            ;HAS SECTOR FRACTION REACHED 77?
006676  001310                          BNE     MRT4D           ;NO

                                        ;CHECK FOR END OF ONE SECTOR OR BEGINNING OF NEXT

006700  010203                  11$:    MOV     R2,R3
006702  042703  177700                  BIC     #177700,R3      ;CHECK SECTOR FRACTION
006706  022703  000077                  CMP     #77,R3          ;END OF SECTOR?
006712  001402                          BEQ     3$              ;YES
006714  000137  006414                  JMP     MRT4B           ;NO, BEGINNING OF NEXT
006720  012737  000012  001206  3$:     MOV     #10.,REPT1      ;SETUP LOOP TO FINISH
006726  012737  000001  001204          MOV     #1,REPT         ;THIS SECTOR
006734  052737  002000  001170          BIS     #BIT10,ONCEE    ;REPT HAS GONE TO ZERO FOR THIS SECTOR
006742  000137  006526                  JMP     MRT4E           ;LOOP

006746  012737  000010  001204  MRT4F:  MOV     #8.,REPT
006754  104422                  1$:     MRCLK                   ;CLOCK MR WITH AN 11 AND A 1
006756  017700  172136                  MOV     @RSLA,BAD       ;GET RSLA
006762  010201                          MOV     R2,GOOD         ;R2 SHOULD=7777
006764  020100                          CMP     GOOD,BAD        ;IS RSLA CORRECT-END OF DISK?
006766  001401                          BEQ     2$              ;YES
006770  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS (7777)
006772  005337  001204          2$:     DEC     REPT            ;FINISH
006776  001366                          BNE     1$              ;LOOP
```

```
2005                                    ;SECTOR AND FRACTION IS = TO 7777 TO INDICATE LAST WORD ON THIS TRACK
2006                                    ;RSLA SHOULD EQUAL 7700 ON ANOTHER MAINT CLOCK.
2007
2008  007000  104422          MRT4G:    MRCLK                   ;CLOCK MR WITH AN 11 AND A 1
2009  007002  017700  172112            MOV     @RSLA,BAD       ;GET RSLA
2010  007006  012701  007700            MOV     #7700,GOOD      ;GET CORRECT ANS
2011  007012  020100                    CMP     GOOD,BAD        ;IS RSLA CORRECT?
2012  007014  001401                    BEQ     1$              ;YES
2013  007016  104000                    HLT                     ;RSLA=BAD GOOD=CORRECT ANS
2014  007020  104430          1$:       MRIND                   ;ISSUE AN INDEX PULSE TO
2015                                                            ;CLEAR THE DRIVE
2016  007022  017700  172072            MOV     @RSLA,BAD       ;GET RSLA
2017  007026  005001                    CLR     GOOD            ;GET CORRECT ANS
2018  007030  020100                    CMP     GOOD,BAD        ;IS RSLA CORRECT?
2019  007032  001401                    BEQ     2$              ;YES
2020  007034  104000                    HLT                     ;RSLA=BAD GOOD=CORRECT ANS
2021  007036  104420          2$:       MRCK                    ;CHECK MR REG
2022  007040  022701                    22701                   ;TO EQUAL 22701
2023  007042  104000                    HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                ;*****************************************************************************
                                ;TEST 42                    ILLEGAL FUNCTION TEST
                                ;*****************************************************************************
007044  104400                  TST42:  SCOPE

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

007046  104414                  MRILF:  CLRDK                   ;CLEAR ALL THE DRIVE REGISTERS
007050  042737  000040  001170          BIC     #BIT5,ONCEE     ;CLEAR CLOCK CNT FLAG
007056  032737  000002  001170          BIT     #BIT1,ONCEE     ;WAS THERE AN ERROR
007064  001002                          BNE     MRLF1           ;YES DO NOT CHANGE "ILF" CODE
007066  012702  000003                  MOV     #3,R2           ;SETUP FIRST "ILF" CODE
                                ;PUT DRIVE IN MAINTENANCE MODE

007072  104416                  MRLF1:  MRDMD                   ;PUT DRIVE INTO MAINT MODE
007074  104420                          MRCK                    ;CHECK MR REG TO
007076  022701                          22701                   ;EQUAL 22701
007100  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)

                                ;ASSERT A MAINTENANCE MODE DISK "INDEX" PULSE

007102  104430                          MRIND
007104  010277  171770          MRLF2:  MOV     R2,@RSCS1       ;SEND "ILF" WITH THE "GO" BIT
007110  017700  171776                  MOV     @RSDS,BAD       ;GET DRIVE STATUS REG
007114  012701  150600                  MOV     #150600,GOOD    ;GET CORRECT ANS
007120  020100                          CMP     GOOD,BAD        ;IS RSDS CORRECT?
007122  001440                          BEQ     1$              ;YES
007124  104402  007130                  TYPE    .+2             ;.ASCIZ <15><12>"ILLEGAL FUNCTION CODE SENT TO DRIVE= "
007200  010237  001216                  MOV     R2,WORK         ;GET FUNCTION CODE
007204  013746  001216                  MOV     WORK,-(6)       ;PUT WORK ON STACK
007210  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
007212  052737  000002  001170          BIS     #BIT1,ONCEE     ;SET ERROR BIT SO ILLEGAL FUN DOESN'T CHANGE
007220  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
007222  104040                          HLT     !DS

007224  042737  000002  001170  1$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR FLAG
007232  017700  171656                  MOV     @RSER,BAD       ;GET RSER
007236  012701  000001                  MOV     #1,GOOD         ;GET CORRECT ANS
007242  020100                          CMP     GOOD,BAD        ;DID "ILF" SET IN RSER
007244  001404                          BEQ     2$              ;YES
007246  052737  000002  001170          BIS     #BIT1,ONCEE     ;SET ERROR BIT
007254  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
007256  042737  000002  001170  2$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR FLAG
```

```
2074                                        ;CLEAR THE DRIVE FOR THE NEXT "ILF" CODE PASS
2075  007264  104414                 MRCILF: CLRDK                   ;CLEAR ERRORS
2076  007266  017700  171620                 MOV     @RSDS,BAD      ;GET RSDS REG
2077  007272  012701  010600                 MOV     #10600,GOOD    ;GET CORRECT ANS
2078  007276  020100                         CMP     GOOD,BAD       ;DID "ATA" AND "ERR" CLEAR IN RSDS?
2079  007300  001435                         BEQ     1$             ;YES
2080  007302  104402  007306               1 TYPE    ,.+2           ;.ASCIZ <15><12>"ATA AND ERR IN RSDS SHOULD CLEAR WITH I
2081  007364  052737  000002  001170         BIS     #BIT1,ONCEE
2082  007372  104000                         HLT                     ;RSDS=BAD GOOD=CORRECT ANS
2083  007374  042737  000002  001170  1$:    BIC     #BIT1,ONCEE    ;CLEAR ERROR FLAG
2084  007402  017700  171506                 MOV     @RSER,BAD      ;GET RSER
2085  007406  005001                         CLR     GOOD           ;GET CORRECT ANS
2086  007410  020100                         CMP     GOOD,BAD       ;DID ILF CLEAR IN RSER
2087  007412  001431                         BEQ     2$             ;YES
2088  007414  052737  000002  001170         BIS     #BIT1,ONCEE    ;SET ERROR BIT
2089  007422  104402  007426               1 TYPE    ,.+2           ;.ASCIZ <15><12>"ILF IN RSER SHOULD CLEAR WITH INIT"
2090  007474  104000                         HLT                     ;RSER=BAD GOOD=CORRECT ANS
2091  007476  042737  000002  001170  2$:    BIC     #BIT1,ONCEE    ;CLEAR ERROR BIT
2092                                        ;GET NEXT ILLEGAL FUNCTION COE
2093
2094  007504  062702  000002         MRLF3: ADD     #2,R2           ;UPDATE ILF
2095  007510  022702  000011                CMP     #11,R2         ;IS THIS A ILF CODE
2096  007514  001773                         BEQ     MRLF3          ;NO-UPDATE IT
2097  007516  022702  000021                CMP     #21,R2
2098  007522  001770                         BEQ     MRLF3
2099  007524  022702  000031                CMP     #31,R2
2100  007530  001765                         BEQ     MRLF3
2101  007532  022702  000051                CMP     #51,R2
2102  007536  001762                         BEQ     MRLF3
2103  007540  022702  000061                CMP     #61,R2
2104  007544  001757                         BEQ     MRLF3
2105  007546  022702  000071                CMP     #71,R2
2106  007552  001754                         BEQ     MRLF3
2107  007554  022702  000101                CMP     #101,R2
2108  007560  001402                         BEQ     ILFDON         ;FINISHED ALL ILF CODES GET OUT
2109  007562  000137  007072                JMP     MRLF1          ;START NEXT ILF FUNCTION
2110  007566                         ILFDON:
```

```
2111                                    ;*****************************************************************************
2112                                    ;TEST 43                    TEST NO-OP CODES 1 AND 21
2113                                    ;*****************************************************************************
2114  007566  104400                    TST43:  SCOPE
2115
2116                                    ;MODULE TESTED M7759
2117  007570  104414                    MROP:   CLRDK                   ;CLEAR ALL DRIVE REGISTERS
2118  007572  042737  000004  001170            BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
2119  007600  104416                            MRDMD                   ;PUT DRIVE INTO MAINT MODE
2120  007602  104420                            MRCK                    ;CHECK MR REG TO
2121  007604  022701                            22701                   ;EQUAL 22701
2122  007606  104424                            MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
2123  007610  032737  000010  001170            BIT     #BIT3,ONCEE     ;TESTING CODE 1
2124  007616  001031                            BNE     3$              ;NO CODE 21
2125  007620  012777  000001  171252            MOV     #1,@RSCS1       ;LOAD NO-OP FUNCTION
2126  007626  012737  000001  001216            MOV     #1,WORK         ;LOAD NO-OP FUNCTION
2127  007634  005777  171254                    TST     @RSER           ;ANY ERRORS
2128  007640  001403                            BEQ     1$              ;NO
2129  007642  004737  022026                    JSR     PC,NOPERR       ;TYPE IT
2130  007646  104040                            HLT     !D$             ;TYPE ERROR
2131  007650  022777  010600  171234  1$:       CMP     #10600,@RSDS    ;IS RSDS CORRECT
2132  007656  001403                            BEQ     2$              ;YES
2133  007660  004737  022026                    JSR     PC,NOPERR       ;RSDS SHOULD
2134  007664  104040                            HLT     !D$             ;EQUAL 10600
2135  007666  042737  000004  001170  2$:       BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
2136
2137                                    ;TEST NO-OP FUNCTION CODE 21
2138
2139  007674  052737  000010  001170            BIS     #BIT3,ONCEE     ;TEST TESTING CODE 21 FLAG
2140  007702  012737  000021  001216  3$:       MOV     #21,WORK        ;LOAD CODE 21
2141  007710  012777  000021  171162            MOV     #21,@RSCS1      ;LOAD FUNCTION
2142  007716  005777  171172                    TST     @RSER           ;ANY ERRORS?
2143  007722  001403                            BEQ     4$              ;NO
2144  007724  004737  022026                    JSR     PC,NOPERR       ;YES, TYPE ERROR
2145  007730  104040                            HLT     !D$             ;ERROR DURING NO-OP FUNCTION
2146  007732  022777  010600  171152  4$:       CMP     #10600,@RSDS    ;IS RSDS CORRECT
2147  007740  001403                            BEQ     5$              ;YES
2148  007742  004737  022026                    JSR     PC,NOPERR       ;TYPE ERROR
2149  007746  104040                            HLT     !D$             ;RSDS SHOULD=10600
2150  007750  042737  000014  001170  5$:       BIC     #14,ONCEE       ;CLEAR TEST BITS
```

```
                                ;*****************************************************************************
                                ;TEST 44                    TEST NO-OP FUNCTION WITH ERROR BITS SET
                                ;*****************************************************************************
007756  104400                  TST44:  SCOPE

                                ;MODULE TESTED M7759
007760  104414                  MROPER: CLRDK                   ;CLEAR ALL REGISTERS
007762  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
007764  104420                          MRCK                    ;CHECK MR REG
007766  022701                          22701                   ;TO EQUAL 22701
007770  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
007772  104430                          MRIND                   ;SEND INDEX PULSE

007774  012777  177777  171112          MOV     #-1,@RSER       ;LOAD RSER WITH ERRORS
010002  013701  001166                  MOV     UNCMP,GOOD      ;GET DRIVE UNDER TEST
010006  042701  177400                  BIC     #177400,GOOD
010012  017700  171100                  MOV     @RSAS,BAD       ;GET RSAS REG
010016  020100                          CMP     GOOD,BAD        ;DID ATA BIT SET CAUSED BY ERROR
010020  001427                          BEQ     1$              ;YES
010022  104402  010026                  TYPE    ,.+2            ;.ASCIZ <15><12>"SET ERRORS IN RSER-RSAS IS INCORRECT"
010076  104000                          HLT                     ;RSAS=BAD GOOD=CORRECT ANS
010100  012737  000001  001216  1$:     MOV     #1,WORK         ;SETUP FOR NO-OP CODE 1
010106  032737  000010  001170          BIT     #BIT3,ONCEE     ;TESTING CODE 2
010114  001004                          BNE     2$              ;YES
010116  012777  000001  170754          MOV     #1,@RSCS1       ;SEND NO-OP CODE 1
010124  000406                          BR      3$              ;CHECK FOR ERRORS
010126  012737  000021  001216  2$:     MOV     #21,WORK        ;SETUP FOR CODE 21
010134  012777  000021  170736          MOV     #21,@RSCS1      ;SENT NO-OP CODE 21
010142  017700  170746          3$:     MOV     @RSER,BAD       ;GET RSER REG
010146  012701  177017                  MOV     #177017,GOOD    ;GET CORRECT ANS
010152  020100                          CMP     GOOD,BAD        ;DID RSER CHANGE WITH NO-OP
010154  001411                          BEQ     4$              ;NO
010156  104402  010162                  TYPE    ,.+2            ;.ASCIZ <15><12>"RSER "
010172  004737  022122                  JSR     PC,CHG
010176  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
010200  017700  170712          4$:     MOV     @RSAS,BAD       ;GET RSAS
010204  013701  001166                  MOV     UNCMP,GOOD      ;GET CORRECT ANS
010210  042701  177400                  BIC     #177400,GOOD    ;CLEAR JUNK
010214  020100                          CMP     GOOD,BAD        ;IS RSAS CORRECT
010216  001411                          BEQ     5$              ;YES
010220  104402  010224                  TYPE    ,.+2            ;.ASCIZ <15><12>"RSAS "
010234  004737  022122                  JSR     PC,CHG          ;TYPE ERROR
010240  104000                          HLT                     ;RSAS=BAD GOOD=CORRECT ANS
010242  017700  170644          5$:     MOV     @RSDS,BAD       ;GET RSDS
010246  012701  150600                  MOV     #150600,GOOD    ;GET CORRECT ANS
010252  020100                          CMP     GOOD,BAD        ;DID RSDS CHANGE
010254  001411                          BEQ     6$              ;NO
010256  104402  010262                  TYPE    ,.+2            ;.ASCIZ <15><12>"RSDS "
010272  004737  022122                  JSR     PC,CHG          ;TYPE ERROR
010276  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
010300  032737  000010  001170  6$:     BIT     #BIT3,ONCEE     ;TESTING CODE 21
010306  001005                          BNE     7$              ;YES, GET OUT
010310  052737  000010  001170          BIS     #BIT3,ONCEE     ;SET CODE 21 FLAG
010316  000137  007760                  JMP     MROPER          ;TEST CODE 21
010322  042737  000010  001170  7$:     BIC     #BIT3,ONCEE     ;DONE CLEAR FLAG AND CONT.
```

```
                                ;*****************************************************************
                                ;TEST 45                    BLOCK SEARCH TEST 1
                                ;*****************************************************************
010330  104400                  TST45:  SCOPE

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

010332  104414          MRSRCH: CLRDK                   ;CLEAR ALL REGISTERS
010334  052737  000040  001170          BIS     #BIT5,ONCEE     ;SET CLOCK FLAG
010342  104416                  MRDMD                   ;PUT DRIVE INTO MAINTENANCE MOE
010344  104420                  MRCK                    ;CHECK MR REG
010346  022701                  22701                   ;TO EQUAL 22701
010350  104424                  MRINT                   ;INIT MR REG (CLEAR MRSP)
010352  104430                  MRIND                   ;CLOCK INDEX PULSE IN RSMR
010354  012777  000003  170526          MOV     #3,@RSDA        ;DO A SEARCH FOR SECTOR 3 OR 41
010362  022777  000000  170536          CMP     #0,@RSDT        ;INTERLEAVED?
010370  001403                  BEQ     4$              ;NO SECTOR 3
010372  012777  000041  170510          MOV     #41,@RSDA       ;YES SECTOR 41
010400  012777  000031  170472  4$:     MOV     #31,@RSCS1      ;LOAD SEARCH COMMAND (M7759)
010406  104426                  DSCK                    ;CHECK RSDS
010410  030400                  30400                   ;TO EQUAL 30400
010412  104000                  HLT                     ;PIP SHOULD BE SET AND DRY SHOULD
                                                        ;BE 0 FOR A DRIVE SEARCH CMD
010414  012737  010643  001204          MOV     #10643,REPT     ;STEP THROUGH 3 SECTORS
010422  104422          1$:     MRCLK                   ;CLOCK MR
010424  104426                  DSCK                    ;RSDS SHOULD NOT
010426  030400                  30400                   ;CHANGE TILL CLOCKING IS COMPLETED
010430  104000                  HLT                     ;TO REACH SECTOR 3
010432  005337  001204                  DEC     REPT            ;KEEP CLOCKING TILL
010436  001371                  BNE     1$              ;SECTOR 3 HAS BEEN REACHED
                                ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
010440  104446                  MCLK1                   ;CLOCK MR REG
010442  104426                  DSCK                    ;CHECK FOR "ATA" AND "DRY"
010444  110600                  110600                  ;TO BE SET IN RSDS FOR
010446  104000                  HLT                     ;SEARCH FUNCTION SHOULD BE COMPLETED
010450  022777  104230  170422          CMP     #104230,@RSCS1  ;SET RSCS1
010456  001401                  BEQ     2$              ;SC IN RSCS1 SHOULD SET BECAUSE OF
010460  104140                  HLT     !DS!AS          ;COMPLETED SEARCH FUNCTION
010462  013777  001164  170426  2$:     MOV     UNITSV,@RSAS    ;CLEAR ATA
010470  005777  170422                  TST     @RSAS           ;DID ATA CLEAR BY WRITING INTO IT?
010474  001401                  BEQ     3$              ;YES
010476  104140                  HLT     !DS!AS          ;RSAS SHOULD=0
010500  022777  004230  170372  3$:     CMP     #4230,@RSCS1    ;DID SC CLEAR BY CLEARING
010506  001401                  BEQ     .+4             ;"ATA" YES
010510  104140                  HLT     !DS!AS          ;NO
```

```
                                ;*************************************************************************
                                ;TEST 46                    BLOCK SEARCH TEST 2
                                ;*************************************************************************
010512  104400                  TST46:  SCOPE

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

010514  104414          MRSRC:  CLRDK                   ;CLEAR ALL REGISTERS
010516  052737  000040  001170          BIS     #BIT5,ONCEE     ;SET CLOCK FLAG
010524  104416                  MRDMD                   ;PUT DRIVE INTO MAINTENANCE MOE
010526  104420                  MRCK                    ;CHECK MR REG
010530  022701                  22701                   ;TO EQUAL 22701
010532  104424                  MRINT                   ;INIT MR REG (CLEAR MRSF)
                                ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
010534  104430                  MRIND
010536  104420                  MRCK                    ;CHECK MR REG TO EQUAL
010540  022701                  22701                   ;22701
010542  104000                  HLT
                                ;STEP THRU RESYNC PERIOD
010544  012737  001000  001204          MOV     #512.,REPT
010552  052737  000040  001170          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
010560  104446          MRRT1:  MCLK1                   ;CLOCK MR REG
010562  104420                  MRCK                    ;CHECK FOR
010564  032711                  32711                   ;CORRECT DATA
010566  104000                  HLT                     ;MR = BAD GOOD = CORRECT DATA
010570  104450                  MCLKO                   ;CLOCK MR REG
010572  104420                  MRCK                    ;CHECK FOR
010574  022701                  22701                   ;CORRECT DATA
010576  104000                  HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
010600  005337  001204                  DEC     REPT            ;FINISH LOOPING
010604  001365                  BNE     MRRT1           ;THROUGH RESYNC PERIOD
                                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE SP = 0
010606  104446                  MCLK1                   ;CLOCK MR REG
010610  104420                  MRCK                    ;MR SHOULD
010612  032311                  32311                   ;EQUALS 32311
010614  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
010616  104450                  MCLKO                   ;CLOCK MR REG
010620  104420                  MRCK                    ;CHECK MR
010622  022301                  22301                   ;TO EQUAL 22301
010624  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
010626  012737  000100  001204          MOV     #100,REPT       ;STEP INTO SECTOR 0
010634  104422          2$:     MRCLK                   ;CLOCK MR REG
010636  005337  001204                  DEC     REPT            ;DO 100 TIMES
010642  001374                  BNE     2$              ;DONE YET? NO BR
010644  012777  000031  170226  4$:     MOV     #31,@RSCS1      ;LOAD SEARCH COMMAND (M7759) FOR SECTOR 0
010652  104426                  DSCK                    ;CHECK RSDS
010654  030400                  30400                   ;TO EQUAL 30400
010656  104000                  HLT                     ;PIP SHOULD BE SET AND DRY SHOULD
                                                        ;BE 0 FOR A DRIVE SEARCH CMD
010660  012737  021506  001204          MOV     #21506,REPT     ;STEP 3 SECTORS BEYOND SECTOR 0
```

```
2310  010666  104422                  1$:     MRCLK                   ;CLOCK MR
2311  010670  104426                          DSCK                    ;RSDS SHOULD NOT
2312  010672  030400                          30400                   ;CHANGE TILL CLOCKING IS COMPLETED
2313  010674  104000                          HLT                     ;TO REACH SECTOR 3
2314  010676  005337  001204                  DEC     REPT            ;KEEP CLOCKING TILL
2315  010702  001371                          BNE     1$              ;SECTOR 3 HAS BEEN REACHED
2316                                  ;ASSERT INDEX PULSE TO SIMULATE THE BEGINNING OF THE NEXT REVOLUTION
2317  010704  104430                          MRIND
2318  010706  104420                          MRCK                    ;CHECK MR REG TO EQUAL
2319  010710  022701                          22701                   ;22701
2320  010712  104000                          HLT
2321
2322                                  ;STEP THRU RESYNC PERIOD
2323
2324  010714  012737  001000  001204          MOV     #512.,REPT
2325  010722  052737  000040  001170          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
2326  010730  104446          MRWR1:  MCLK1                           ;CLOCK MR REG
2327  010732  104420                  MRCK                            ;CHECK FOR
2328  010734  032711                  32711                           ;CORRECT DATA
2329  010736  104000                  HLT                             ;MR = BAD GOOD = CORRECT DATA
2330  010740  104450                  MCLK0                           ;CLOCK MR REG
2331  010742  104420                  MRCK                            ;CHECK FOR
2332  010744  022701                  22701                           ;CORRECT DATA
2333  010746  104000                  HLT                             ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2334  010750  005337  001204                  DEC     REPT            ;FINISH LOOPING
2335  010754  001365                          BNE     MRWR1           ;THROUGH RESYNC PERIOD
2336
2337                                  ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2338                                  ;SP=0 EQUALS SECTOR PULSE
2339  010756  104446                          MCLK1                   ;CLOCK MR REG
2340  010760  104420                          MRCK                    ;MR SHOULD
2341  010762  032311                          32311                   ;EQUAL 32311
2342  010764  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2343  010766  104450                          MCLK0                   ;CLOCK MR REG
2344  010770  104420                          MRCK                    ;CHECK MR
2345  010772  022301                          22301                   ;TO EQUAL 22301
2346  010774  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2347
2348                                  ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
2349  010776  104446                          MCLK1                   ;CLOCK MR REG
2350  011000  104426                          DSCK                    ;CHECK FOR "ATA" AND "DRY"
2351  011002  110600                          110600                  ;TO BE SET IN RSDS FOR
2352  011004  104000                          HLT                     ;SEARCH FUNCTION SHOULD BE COMPLETED
2353  011006  022777  104230  170064          CMP     #104230,@RSCS1  ;SET RSCS1
2354  011014  001401                          BEQ     2$              ;SC IN RSCS1 SHOULD SET BECAUSE OF
2355  011016  104140                          HLT     !DS!AS          ;COMPLETED SEARCH FUNCTION
2356  011020  013777  001164  170070  2$:     MOV     UNITSV,@RSAS    ;CLEAR ATA
2357  011026  005777  170064                  TST     @RSAS           ;DID ATA CLEAR BY WRITING INTO IT?
2358  011032  001401                          BEQ     3$              ;YES
2359  011034  104140                          HLT     !DS!AS          ;RSAS SHOULD=0
2360  011036  022777  004230  170034  3$:     CMP     #4230,@RSCS1    ;DID SC CLEAR BY CLEARING
2361  011044  001401                          BEQ     .+4             ;"ATA" YES
2362  011046  104140                          HLT     !DS!AS  ;NO
```

```
                                ;*************************************************************************
                                ;TEST 47                 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSCS1)
                                ;*************************************************************************
011050  104400          TST47:  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

011052  104414  RMRC1:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
011054  042737  000040  001170        BIC     #BIT5,ONCEE     ;CLEAR CLK CNT FLAG
011062  104416          MRDMD                   ;PUT DRIVE INTO MAINT MODE
011064  104420          MRCK                    ;CHECK MR REG TO
011066  022701          22701                   ;EQUAL 22701
011070  104424          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
011072  012777  000001  170010        MOV     #1,@RSDA        ;LOAD RSDA
011100  012777  000031  167772        MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
011106  104426          DSCK                    ;CHECK RSDS
011110  030400          30400                   ;TO EQUAL 30400
011112  104000          HLT                     ;DRY IN RSDS SHOULD BE
                                                ;CLEARED FOR DRIVE WAS
                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                ;RSDS=BAD GOOD=CORRECT ANS
011114  012777  000011  167756        MOV     #11,@RSCS1      ;LOAD A CLEAR FUNCTION
                                                ;THIS SHOULD CAUSE AN RMR
                                                ;ERROR FOR DRIVE WAS BUSY
                                                ;WHEN CLEAR COMMAND WAS GIVEN
011122  017700  167766          MOV     @RSER,BAD       ;GET RSER REG
011126  012701  000004          MOV     #4,GOOD         ;GET CORRECT ANS
011132  020100          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
011134  001410          BEQ     1$              ;YES
011136  104402  022173          TYPE    ,TRMR
011142  104402  011146          TYPE    ,.+2            ;.ASCIZ "RSCS1"
011154  104000          HLT                     ;RSER=BAD GOOD=CORRECT ANS
011156  104426  1$:     DSCK                    ;CHECK RSDS TO
011160  150600          150600                  ;EQUAL 150600
011162  104000          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
011164  022777  104230  167706        CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
011172  001401          BEQ     2$              ;YES
011174  104040          HLT     !DS             ;RSCS1 SHOULD=104230
                                                ;RSDS SHOULD=150600
                                                ;RSER SHOULD=4
011176  022777  000001  167704  2$:     CMP     #1,@RSDA        ;DID CLR CLEAR RSDA
011204  001401          BEQ     4$              ;NO
011206  104004          HLT     !DA             ;RSDA SHOULD=1
011210  104414  4$:     CLRDK                   ;CLEAR ALL REGISTERS
011212  005777  167676          TST     @RSER           ;RSER SHOULD CLEAR
011216  001401          BEQ     3$              ;RSER OK
011220  104040          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                ;CLEAR BIT WAS LOADED IN RSCS2
011222  022777  004200  167650  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
011230  001401          BEQ     .+4             ;CLEAR BIT WAS LOADED IN RSCS2
011232  104040          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
2416                                    ;*************************************************************************
2417                                    ;TEST 50                DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSDA)
2418                                    ;*************************************************************************
2419  011234  104400                    TST50:  SCOPE
2420
2421                                    ;MODULE TESTED M7755 M7759 M7770
2422                                    ;RMR ERROR IS CAUSED BY WRITTING INTO RSDA WHILE DOING A BLOCK SEARCH FUNCTION
2423
2424  011236  104414            RMRC2:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
2425  011240  104416                    MRDMD                   ;PUT DRIVE INTO MAINT MODE
2426  011242  104420                    MRCK                    ;CHECK MR REG TO
2427  011244  022701                    22701                   ;EQUAL 22701
2428  011246  104424                    MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
2429  011250  012777  000001  167632    MOV     #1,@RSDA        ;LOAD RSDA
2430  011256  012777  000031  167614    MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
2431  011264  104426                    DSCK                    ;CHECK RSDS
2432  011266  030400                    30400                   ;TO EQUAL 30400
2433  011270  104000                    HLT                     ;DRY IN RSDS SHOULD BE
2434                                                            ;CLEARED FOR DRIVE WAS
2435                                                            ;ISSURED A BLOCK SEARCH FUNCTION
2436                                                            ;RSDS=BAD GOOD=CORRECT ANS
2437  011272  005077  167612            CLR     @RSDA           ;MODIFY RSDA
2438                                                            ;THIS SHOULD CAUSE AN RMR
2439                                                            ;ERROR FOR DRIVE WAS BUSY
2440                                                            ;WHEN COMMAND WAS GIVEN
2441  011276  017700  167612            MOV     @RSER,BAD       ;GET RSER REG
2442  011302  012701  000004            MOV     #4,GOOD         ;GET CORRECT ANS
2443  011306  020100                    CMP     GOOD,BAD        ;DID RMR SET IN RSER?
2444  011310  001410                    BEQ     1$              ;YES
2445  011312  104402  022173            TYPE    ,TRMR
2446  011316  104402  011322            TYPE    ,.+2            ;.ASCIZ "RSDA"
2447  011330  104000                    HLT                     ;RSER=BAD GOOD=CORRECT ANS
2448  011332  104426            1$:     DSCK                    ;CHECK RSDS TO
2449  011334  150600                    150600                  ;EQUAL 150600
2450  011336  104000                    HLT                     ;RSDS=BAD GOOD=CORRECT ANS
2451  011340  022777  104230  167532    CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
2452  011346  001401                    BEQ     2$              ;YES
2453  011350  104040                    HLT     !DS             ;RSCS1 SHOULD=104230
2454                                                            ;RSDS SHOULD=150600
2455                                                            ;RSER SHOULD=4
2456  011352  022777  000001  167530    2$:     CMP     #1,@RSDA        ;DID CLR CLEAR RSDA
2457  011360  001401                    BEQ     4$              ;NO
2458  011362  104004                    HLT     !DA             ;RSDA SHOULD=1
2459  011364  104414            4$:     CLRDK                   ;CLEAR ALL REGISTERS
2460  011366  005777  167522            TST     @RSER           ;RSER SHOULD CLEAR
2461  011372  001401                    BEQ     3$              ;RSER OK
2462  011374  104040                    HLT     !DS             ;RSER SHOULD=0 FOR THE
2463                                                            ;CLEAR BIT WAS LOADED IN RSCS2
2464  011376  022777  004200  167474    3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
2465  011404  001401                    BEQ     .+4             ;CLEAR BIT WAS LOADED IN RSCS2
2466  011406  104040                    HLT     !DS             ;RSCS1 SHOULD=4200
2467
```

```
                                ;*****************************************************************************
                                ;TEST 51                    DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)
                                ;*****************************************************************************
011410  104400                  TST51:  SCOPE

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

011412  104414                  RMRC3:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
011414  042737  000040  001170          BIC     #BIT5,ONCEE     ;CLEAR CLOCK COUNT FLAG
011422  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
011424  104420                          MRCK                    ;CHECK MR REG TO
011426  022701                          22701                   ;EQUAL 22701
011430  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
011432  012777  000001  167450          MOV     #1,@RSDA        ;LOAD RSDA
011440  012777  000031  167432          MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
011446  104426                          DSCK                    ;CHECK RSDS
011450  030400                          30400                   ;TO EQUAL 30400
011452  104000                          HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
011454  012777  177777  167432          MOV     #-1,@RSER       ;MODIFY RSER
                                                                ;THIS SHOULD CAUSE AN RMR
                                                                ;ERROR FOR DRIVE WAS BUSY
                                                                ;WHEN COMMAND WAS GIVEN
011462  017700  167426                  MOV     @RSER,BAD       ;GET RSER REG
011466  012701  000004                  MOV     #4,GOOD         ;GET CORRECT ANS
011472  020100                          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
011474  001410                          BEQ     1$              ;YES
011476  104402  022173                  TYPE    ,TRMR
011502  104402  011506                  TYPE    ,.+2
011514  104000                          HLT                     ;.ASCIZ "RSER"
                                                                ;RSER=BAD GOOD=CORRECT ANS
011516  104426                  1$:     DSCK                    ;CHECK RSDS TO
011520  150600                          150600                  ;EQUAL 150600
011522  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
011524  022777  104230  167346          CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
011532  001401                          BEQ     4$              ;YES
011534  104040                          HLT     !DS             ;RSCS1 SHOULD=104230
                                                                ;RSDS SHOULD=150600
                                                                ;RSER SHOULD=4
011536  104414                  4$:     CLRDK                   ;CLEAR ALL REGISTERS
011540  005777  167350                  TST     @RSER           ;RSER SHOULD CLEAR
011544  001401                          BEQ     3$              ;RSER OK
011546  104040                          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LOADED IN RSCS2
011550  022777  004200  167322  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
011556  001401                          BEQ     .+4             ;CLEAR BIT WAS LOADER IN RSCS2
011560  104040                          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                ;*************************************************************************
                                ;TEST 52                 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSAS)
                                ;*************************************************************************
011562  104400                  TST52:  SCOPE

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

011564  104414                  RMRC4:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
011566  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
011570  104420                          MRCK                    ;CHECK MR REG TO
011572  022701                          22701                   ;EQUAL 22701
011574  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
011576  012777  000001  167304          MOV     #1,@RSDA        ;LOAD RSDA
011604  012777  000031  167266          MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
011612  104426                          DSCK                    ;CHECK RSDS
011614  030400                          30400                   ;TO EQUAL 30400
011616  104000                          HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
011620  005077  167272                  CLR     @RSAS           ;WRITE INTO ATTENTION SUMMARY REGISTER
                                                                ;SHOULD BE NO RMR ERROR BECAUSE
                                                                ;WRITING RSAS IN ALLOWED ANYTIME.
011624  017700  167264                  MOV     @RSER,BAD       ;GET RSER REG
011630  012701  000000                  MOV     #0,GOOD         ;GET CORRECT ANS
011634  020100                          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
011636  001435                          BEQ     1$              ;NO
011640  104402  011644                  TYPE    ,.+2            ;.ASCIZ <15><12>"RMR ERROR SHOULD NOT SET WHILE WRITING
011730  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
011732  104426                  1$:     DSCK                    ;CHECK RSDS TO
011734  030400                          30400                   ;EQUAL 30400
011736  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
011740  022777  004231  167132          CMP     #4231,@RSCS1    ;DID CORRECT BITS SET IN RSCS1
011746  001401                          BEQ     4$              ;YES
011750  104040                          HLT     !DS             ;RSCS1 SHOULD=4231
                                                                ;RSDS SHOULD=30400
                                                                ;RSER SHOULD=0
011752  104414                  4$:     CLRDK                   ;CLEAR ALL REGISTEDS
011754  005777  167134                  TST     @RSER           ;RSER SHOULD CLEAR
011760  001401                          BEQ     3$              ;RSER OK
011762  104040                          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LOADED IN RSCS2
011764  022777  004200  167106  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
011772  001401                          BEQ     .+4             ;CLEAR BIT WAS LOADER IN RSCS2
011774  104040                          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                ;*****************************************************************
                                ;TEST 53                    DRIVE SELECT TEST
                                ;*****************************************************************
011776  104400                  TST53:  SCOPE

                                ;MODULE TESTED: M7755
                                ;THE PROGRAM LOADS A DRIVE REGISTER, OF THE DRIVE UNDER TEST, TO ALL ONES.
                                ;THE PROGRAM THEN FINDS A NON-EXISTENT DRIVE AND TRIES TO LOAD ITS
                                ;REGISTER WITH ALL ZEROS. THIS SHOULD CAUSE "NED" TO
                                ;SET IN RSCS2. THE PROGRAM RE-SELECTS THE DRIVE UNDER TEST AND CHECKS
                                ;ITS REGISTER TO SEE IF IT WAS MODIFIED. IT SHOULD CONTAIN ALL ONES.
                                ;CHECK UNIT NO. COMPARATOR, E19-M7755 IF TEST FAILS

012000  104414                  MRDSEL: CLRDK                   ;CLEAR ALL REGISTERS
012002  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
012004  104420                          MRCK                    ;CHECK MAINT REG
012006  022701                          22701                   ;TO EQUAL 22701
012010  104424                          MRINT                   ;INITIALIZE MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
012012  012777  177777  167070          MOV     #-1,@RSDA       ;LOAD DISK ADDR REG OF DRIVE UNDER TEST

                                ;SEARCH FOR NON EXISTENT DRIVES

012020  012737  000401  001216          MOV     #401,WORK
012026  005001                          CLR     GOOD
012030  010177  167046          1$:     MOV     GOOD,@RSCS2     ;LOAD UNIT NO
012034  005777  167054                  TST     @RSER           ;IS THIS A NED?
012040  032777  010000  167034          BIT     #BIT12,@RSCS2   ;IS THIS A NED?
012046  001005                          BNE     2$              ;FOUND NED
012050  005201                          INC     GOOD            ;UPDATE UNIT NUMBER
012052  006137  001216                  ROL     WORK            ;KEEP LOOKING FOR NED
012056  103460                          BCS     NEDDON          ;COULD NOT FIND ANY NON EXISTENT DRIVES
012060  000763                          BR      1$              ;LOOK FOR NED
012062  012777  004000  167010  2$:     MOV     #4000,@RSCS1    ;CLEAR NED
012070  010137  001222                  MOV     GOOD,WORK1      ;SAVE NED NUMBER
012074  010177  167002                  MOV     GOOD,@RSCS2     ;LOAD UNIT # OF NED INTO RSCS2
012100  005077  167004                  CLR     @RSDA           ;WRITE INTO A NON EXISTENT DRIVE REG
                                                                ;THIS SHOULD CAUSE NED TO
                                                                ;SET IN RSCS2
012104  017700  166772                  MOV     @RSCS2,BAD      ;GET RSCS2
012110  052701  010100                  BIS     #10100,GOOD     ;PUT CORRECT ANS IN GOOD
                                                                ;BY SETTING NED AND IR
012114  020100                          CMP     GOOD,BAD        ;IS RSCS2 CORRECT?
012116  001401                          BEQ     .+4             ;YES
012120  104000                          HLT                     ;RSCS2=BAD GOOD=CORRECT ANS

012122  022777  160200  166750          CMP     #160200,@RSCS1  ;IS CS1 CORRECT
012130  001401                          BEQ     .+4             ;YES
012132  104004                          HLT     !DA             ;TRE SHOULD BE SET IN CS1 BECAUSE
                                                                ;OF NED ERROR IN RSCS2
                                                                ;RSCS1 SHOULD=160200
```

```
2616  012134  005777  166756                    TST     @RSAS           ;DID ANY ATTENTION BITS SET?
2617  012140  001401                            BEQ     .+4             ;NO
2618  012142  104100                            HLT     !AS             ;NO ATTENTION BITS SHOULD BE SET
2619  012144  112777  000100  166762            MOVB    #100,@RSCS1B    ;CLEAR TRE
2620  012152  032777  010000  166722            BIT     #NED,@RSCS2     ;DID NED CLEAR
2621  012160  001401                            BEQ     .+4             ;YES
2622  012162  104040                            HLT     !DS             ;NED DID NOT CLEAR IN RSCS2
2623                                                                    ;BY CLEARING TRE BIT IN RSCS1
2624  012164  013777  001162  166710            MOV     UNNUM,@RSCS2    ;LOAD CORRECT UNIT NUMBER
2625  012172  022777  177777  166710            CMP     #-1,@RSDA       ;DID RSDA GET MODIFIED
2626                                                                    ;WHILE WRITING INTO A NON
2627                                                                    ;EXISTENT DRIVE?
2628  012200  001443                            BEQ     NNDD            ;NO
2629  012202  104004                            HLT     !DA             ;RSDA SHOULD= -1
2630  012204  013700  001222                    MOV     WORK1,BAD       ;IT GOT MODIFIED WHILE WRITING
2631  012210  013701  001162                    MOV     UNNUM,GOOD      ;INTO A NED
2632  012214  104000                            HLT                     ;GOOD=DRIVE UNDER TEST
2633  012216  000434                            BR      NNDD            ;BAD=NON EXISTENT DRIVE THAT WAS
2634                                                                    ;IN RSCS2 WHEN RSDA GOT MODIFIED
2635  012220  032737  010000  001170  NEDDON:   BIT     #BIT12,ONCEE    ;WAS THIS TYPED BEFORE?
2636  012226  001030                            BNE     NNDD            ;YES
2637  012230  104402  012234                    TYPE    ,.+2            ;.ASCIZ <15><12>"COULD NOT FIND A NON-EXISTENT DRIVE"
2638  012302  052737  010000  001170            BIS     #BIT12,ONCEE    ;SET TYPED MESSAGE FLAG
2639  012310                            NNDD:
```

```
                                ;*****************************************************************************
                                ;TEST 54                    MAINTENANCE MODE WRITE TEST
                                ;*****************************************************************************
012310  104400                  TST54:  SCOPE

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

012312  012737  001602  001144          MOV     #1602,FLAG2     ;SET TEST FLAG
012320  104414  MRWRT:  CLRDK           ;CLEAR DRIVE REGISTERS
012322  012737  000040  001170          MOV     #40,ONCEE       ;SETUP TEST FLAGS
012330  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
012332  104420                          MRCK                    ;CHECK MR REG
012334  022701                          22701                   ;TO EQUAL 22701
012336  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES

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

012340  012702  030114                  MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
012344  005022                          CLR     (R2)+           ;CLEAR 1ST LOCATION
012346  012722  177777                  MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
012352  005003                          CLR     R3              ;CLEAR WORK LOC TO GENERATE
012354  000261                          SEC                     ;A PATTERN OF FLOATING ONES
012356  006103  1$:    ROL     R3      ;GET PATTERN
012360  103402                          BCS     2$              ;DONE GET OUT
012362  010322                          MOV     R3,(R2)+        ;FILL BUFFER
012364  000774                          BR      1$              ;CONT
012366  012703  000056  2$:    MOV     #46.,R3         ;FILL REMAINING PORTION OF

012372  012704  146314                  MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
012376  010422  3$:    MOV     R4,(R2)+        ;LOAD BUFFER
012400  005303                          DEC     R3              ;DONE YET?
012402  001375                          BNE     3$              ;NO

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

012404  012777  030114  166474          MOV     #INBUF,@RSBA    ;LOAD BUS ADDR REG
012412  012777  177700  166464          MOV     #177700,@RSWC   ;LOAD WORD COUNT REG
012420  012777  000061  166452          MOV     #61,@RSCS1      ;LOAD WRITE COMMAND
012426  104454                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
012430  104220                          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
012432  104456                          SPASS                   ;CLOCK MR REG SP = 1
```

```
                                ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
012434  104430                          MRIND
012436  104420                          MRCK                    ;CHECK MR REG TO EQUAL
012440  020501                          20501                   ;20501 FOR A
012442  104000                          HLT                     ;WRITE COMD HAS BEEN ISSUED

                                ;STEP THRU RESYNC PERIOD

012444  012737  001000  001204          MOV     #512.,REPT
012452  052737  000040  001170          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
012460  104446                  MRWRT1: MCLK1                   ;CLOCK MR REG
012462  104420                          MRCK                    ;CHECK FOR
012464  030511                          30511                   ;CORRECT DATA
012466  104000                          HLT                     ;MR = BAD GOOD = CORRECT DATA
012470  104450                          MCLK0                   ;CLOCK MR REG
012472  104420                          MRCK                    ;CHECK FOR
012474  020501                          20501                   ;CORRECT DATA
012476  104000                          HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
012500  005337  001204                  DEC     REPT            ;FINISH LOOPING
012504  001365                          BNE     MRWRT1          ;THROUGH RESYNC PERIOD

                                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                ;SP=0 EQUALS SECTOR PULSE
012506  104446                          MCLK1                   ;CLOCK MR REG
012510  104420                          MRCK                    ;MR SHOULD
012512  030111                          30111                   ;EQUAL 30111
012514  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
012516  104450                          MCLK0                   ;CLOCK MR REG
012520  104420                          MRCK                    ;CHECK MR
012522  020101                          20101                   ;TO EQUAL 20101
012524  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS

                                ;PERFORM 63 MAINT CLOCK OPERATIONS--WRITING PREAMBLE

012526  012737  000077  001204          MOV     #63.,REPT
012534  104446                  MRWRT2: MCLK1                   ;CLOCK MR REG
012536  104420                          MRCK                    ;CHECK MR REG
012540  031511                          31511                   ;TO EQUAL 31511
012542  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
012544  104450                          MCLK0                   ;CLOCK MR REG
012546  104420                          MRCK                    ;CHECK MR REG
012550  021501                          21501                   ;TO EQUAL 21501
012552  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
012554  005337  001204                  DEC     REPT            ;DONE YET
012560  001365                          BNE     MRWRT2          ;NO LOOP
```

```
                                        ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN
012562  104446                                MCLK1                   ;CLOCK MR REG
012564  104420                                MRCK                    ;CHECK MR REG
012566  131511                                131511                  ;TO EQUAL 131511
012570  104000                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
012572  104450                                MCLK0                   ;CLOCK MR REG
012574  104420                                MRCK                    ;MR REG SHOULD
012576  025501                                25501                   ;EQUAL 25501
012600  104000                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
012602  104446                                MCLK1                   ;CLOCK MR REG
012604  104420                                MRCK                    ;MR SHOULD EQUAL
012606  135511                                135511                  ;35511
012610  104000                                HLT
                                        ;PERFORM NEXT STEP 8 TIMES TO FINISH WRITING PREAMBLE
012612  012737  000010  001204                MOV     #10,REPT
012620  104450                        MRWRT3: MCLK0                   ;CLOCK MR REG
012622  104420                                MRCK                    ;CHECK MR REG
012624  025501                                25501                   ;TO EQUAL 25501
012626  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
012630  104446                                MCLK1                   ;CLOCK MR REG
012632  104420                                MRCK                    ;CHECK MR REG
012634  135511                                135511                  ;TO EQUAL 135511
012636  104000                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
012640  005337  001204                        DEC     REPT            ;DONE YES?
012644  001365                                BNE     MRWRT3          ;NO LOOP BACK

                                        ;MOVE DATA WORD INTO RS03 SHIFT REGISTER (M7753)

012646  104450                                MCLK0                   ;CLOCK MR REG
012650  104420                                MRCK                    ;CHECK MR REG
012652  021501                                21501                   ;TO EQUAL 21501
012654  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;ENCODE SYNC 1 (M7751)

012656  104446                                MCLK1                   ;CLOCK MR REG
012660  104420                                MRCK                    ;MR REG SHOULD
012662  123511                                123511                  ;EQUAL 123511
012664  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
012666  104450                                MCLK0                   ;CLOCK MR REG
012670  104420                                MRCK                    ;MR REG SHOULD NOW
012672  033501                                33501                   ;EQUAL 33501
012674  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
012676  012705  030114                        MOV     #INBUF,R5       ;GET STARTING ADDR FOR DATA BUFFER
012702  011504                                MOV     (R5),R4         ;GET DATA
```

```
2785  012704  012737  002156  001216          MOV     #1134.,WORK     ;DOING A 1 SECTOR TRANSFER 63 WORDS
2786                                                                    ;18 BITS PER WORD-CLOCK LOOPS
2787                                                                    ;TAKE CARE OF 1 BIT AT A TIME
2788                                                                    ;63 TIMES 18 EQUALS 1134 LOOPS
2789                                                                    ;TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY)
2790  012712  052737  000100  001170          BIS     #BIT6,ONCEE     ;SET 1ST TRANSFER WORD FLAG
2791  012720  104432                  1$:     XBIT                    ;GET 1 BIT OF DATA
2792  012722  104434                          CLKD1                   ;SET MCLK IN RSMR
2793                                                                    ;AND CALCULATE MR REG
2794                                                                    ;FOR CORRECT DATA (MWDB)
2795  012724  104000                          HLT                     ;MR REG NOT CORRECT
2796  012726  104436                          CLKD0                   ;CLEAR MCLK TO
2797                                                                    ;COMPLETE TRANSFER OF THIS BIT
2798                                                                    ;CALCULATE CORRECT ANS FOR
2799                                                                    ;MR REG (MWDB)
2800  012730  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2801  012732  032737  000200  001170          BIT     #BIT7,ONCEE     ;ON LAST WORD YET?
2802  012740  001015                          BNE     2$              ;YES
2803  012742  032737  000400  001170          BIT     #BIT8,ONCEE     ;ON CRC WORD YET?
2804  012750  001040                          BNE     3$              ;YES
2805  012752  005337  001216                  DEC     WORK            ;DONE WITH 63 WORDS?
2806  012756  001360                          BNE     1$              ;NO
2807
2808  012760  052737  000200  001170          BIS     #BIT7,ONCEE     ;SET LAST WORD FLAG
2809  012766  012737  000023  001216          MOV     #19.,WORK       ;SET UP TO TRANSFER LAST WORD
2810  012774  005337  001216          2$:     DEC     WORK            ;DONE YET?
2811  013000  001347                          BNE     1$              ;NO
2812  013002  052737  000400  001170          BIS     #BIT8,ONCEE     ;SET TRANSFERRING CRC WORD
2813  013010  042737  000200  001170          BIC     #BIT7,ONCEE     ;CLEAR LAST WORD FLAG
2814  013016  004737  024430                  JSR     PC,GENCRC       ;GENERATE CRC WORD
2815                                                                    ;AND LEAVE IN "WORK"
2816  013022  012702  030114                  MOV     #INBUF,R2       ;GO TO END
2817  013026  062702  000200                  ADD     #200,R2         ;OF DATA BUFFER
2818  013032  013712  001216                  MOV     WORK,@R2        ;LOAD CRC WORD
2819  013036  010205                          MOV     R2,R5           ;RESET POINTER FOR
2820  013040  162705  000002                  SUB     #2,R5           ;R5 FOR CRC WD
2821  013044  012737  000023  001216          MOV     #19.,WORK       ;SETUP TO XFER CRC
2822  013052  005337  001216          3$:     DEC     WORK            ;DONE YET
2823  013056  001320                          BNE     1$              ;NO
2824
2825                          ;EBL SHOULD NOW ASSERT
2826
2827  013060  104446                          MCLK1                   ;CLOCK MR REG TO STOP THROUGH
2828                                                                    ;THE RS03 SECTOR DEAD BAND AREA
2829  013062  104420                          MRCK                    ;CHECK MR REG
2830  013064  113511                          113511                  ;TO EQUAL 113511
2831  013066  104000                          HLT                     ;MR REG=BAD GOOD=CORRECT ANS
```

```
2832                                    ;LOOP 17 TIMES
2833
2834  013070  012737  000017  001204          MOV     #17;REPT
2835  013076  104450                  4$:     MCLKO                   ;CLOCK MR REG
2836  013100  104420                          MRCK                    ;CHECK MR REG
2837  013102  003501                          3501                    ;TO EQUAL 3501
2838  013104  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2839  013106  104446                          MCLK1                   ;CLOCK MR REG
2840  013110  104420                          MRCK                    ;CHECK MR REG
2841  013112  113511                          113511                  ;TO EQUAL 113511
2842  013114  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2843  013116  005337  001204                  DEC     REPT            ;DONE LOOPING YET?
2844  013122  001365                          BNE     4$              ;NO
2845
2846                                    ;FINISH UP
2847  013124  104450                          MCLKO                   ;CLOCK MR REG
2848  013126  104420                          MRCK                    ;CHECK MR REG
2849  013130  003501                          3501                    ;TO EQUAL 3501
2850  013132  104000                          HLT                     ;MR REG=BAD GOOD=CORRECT ANS
2851  013134  104446                          MCLK1.                  ;CLOCK MR REG
2852  013136  104420                          MRCK                    ;CHECK MR REG
2853  013140  111511                          111511                  ;TO EQUAL 111511
2854  013142  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2855  013144  104450                          MCLKO                   ;CLOCK MR REG
2856  013146  104420                          MRCK                    ;CHECK MRE REG
2857  013150  001501                          1501                    ;TO EQUAL 1501
2858  013152  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2859
2860                                    ;TRANSFER SHOULD NOW BE COMPLETE
2861
2862  013154  104446                          MCLK1                   ;CLOCK MR REG
2863  013156  104420                          MRCK                    ;CHECK MR
2864  013160  012711                          12711                   ;REG TO
2865  013162  104000                          HLT                     ;EQUAL 12711
2866  013164  104450                          MCLKO                   ;CLOCK MR REG
2867  013166  104420                          MRCK                    ;CHECK MR REG
2868  013170  002701                          2701                    ;TO
2869  013172  104000                          HLT                     ;EQUAL 2701
2870
2871                                    ;NOW TEST CONTROLLER
2872
2873  013174  005777  165700                  TST     @RSCS1          ;ANY ERRORS?
2874  013200  100001                          BPL     5$              ;NO
2875  013202  104014                          HLT     !DA!WC          ;YES
2876  013204  005777  165674          5$:     TST     @RSWC           ;DID WC GO TO 0
2877  013210  001401                          BEQ     .+4             ;YES
2878  013212  104010                          HLT     !WC             ;WC SHOULD BE = TO 0
2879  013214  022777  000001  165666          CMP     #1,@RSDA        ;DID RSDA INCREMENT TO A 1
2880  013222  001401                          BEQ     .+4             ;YES
2881  013224  104004                          HLT     !DA             ;NO RSDA SHOULD=1
2882  013226  032737  000002  001144          BIT     #BIT1,FLAG2     ;IN MAINT VERIFY TEST?
2883  013234  001002                          BNE     .+6             ;NO
2884  013236  000137  021450                  JMP     @#MRVR2         ;YES, GO TO VERIFY TEST
```

```
                                ;*******************************************************************************
                                ;TEST 55                    MAINTENANCE READ TEST
                                ;*******************************************************************************
013242  104400                  TST55:  SCOPE

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

013244  104414                  MRRD:   CLRDK                   ;CLEAR DRIVE REGISTERS
013246  052737  000040  001170          BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
013254  042737  047716  001170          BIC     #47716,ONCEE    ;CLEAR ALL OTHER FLAG BITS
013262  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
013264  104420                          MRCK                    ;CHECK MR REG
013266  022701                          22701                   ;TO EQUAL 22701
013270  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES

013272  005037  001144                  CLR     FLAG2           ;CLEAR FLAG TEST BITS

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

013276  012702  030114                  MOV     #INBUF,R2       ;GET LOCATION OF INBUF
013302  005022                          CLR     (R2)+           ;CLEAR 1ST LOCATION
013304  012722  177777                  MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
013310  005003                          CLR     R3              ;CLEAR WORK LOC TO GENERATE
013312  000261                          SEC                     ;A PATTERN OF FLOATING ONES
013314  006103                  1$:     ROL     R3              ;GET PATTERN
013316  103402                          BCS     2$              ;DONE GET OUT
013320  010322                          MOV     R3,(R2)+        ;FILL BUFFER
013322  000774                          BR      1$              ;CONT
013324  012703  000056          2$:     MOV     #46.,R3         ;FILL REMAINING PORTION OF
013330  012704  146314                  MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
013334  010422                  3$:     MOV     R4,(R2)+        ;LOAD BUFFER
013336  005303                          DEC     R3              ;DONE YET
013340  001375                          BNE     3$              ;NO

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

```
                                        ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
                                        
013342  012777  030714  165536          MOV     #OUTBUF,@RSBA   ;LOAD BUS ADDR REG
013350  012777  177700  165526          MOV     #177700,@RSWC   ;LOAD WORD COUNT REG
013356  012777  000071  165514          MOV     #71,@RSCS1      ;LOAD READ COMMAND
013364  012702  000100                  MOV     #100,R2         ;CLEAR THE OUTBUF TABLE SO THAT
013370  012703  030714                  MOV     #OUTBUF,R3      ;WHEN THE READ IS FINISHED, WE CAN
013374  005023                  4$:     CLR     (R3)+           ;COMPARE WHAT WE GOT (OUTBUF)
013376  005302                  DEC     R2                      ;WITH WHAT WE EXPECTED (INBUF).
013400  001375                  BNE     4$
013402  104454                  GETSP                           ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
013404  104220                  HLT     !MR                     ;COULD NOT SET SECTOR PULSE (0)
013406  104456                  SPASS                           ;CLOCK MR REG SP = 1
                                        
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
013410  104430                  MRIND
013412  104420                  MRCK                            ;CHECK MR REG TO EQUAL
013414  022601                  22601                           ;22601 FOR A
013416  104000                  HLT                             ;READ COMD
                                        
                                        ;STEP THRU RESYNC PERIOD
                                        
013420  012737  001000  001204          MOV     #512.,REPT
013426  052737  000040  001170          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT
013434  104446          MRRD1:  MCLK1                           ;CLOCK MR REG
013436  104420                  MRCK                            ;CHECK FOR
013440  032611                  32611                           ;CORRECT DATA
013442  104000                  HLT                             ;MR=BAD GOOD=CORRECT DATA
013444  104450                  MCLK0                           ;CLOCK MR REG
013446  104420                  MRCK                            ;CHECK FOR
013450  022601                  22601                           ;CORRECT DATA
013452  104000                  HLT                             ;ERROR WHILE CLOCKING THROUGH RESYNC
013454  005337  001204          DEC     REPT                    ;FINISH LOOPING
013460  001365                  BNE     MRRD1                   ;THROUGH RESYNC PERIOD
                                        
                                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                ;SP=0 EQUALS SECTOR PULSE
013462  104446                  MCLK1                           ;CLOCK MR REG
013464  104420                  MRCK                            ;MR SHOULD
013466  032211                  32211                           ;EQUAL 32211
013470  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS
013472  104450                  MCLK0                           ;CLOCK MR REG
013474  104420                  MRCK                            ;CHECK MR
013476  022201                  22201                           ;TO EQUAL 22201
013500  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS
```

```
2979                                    ;PERFORM 71 MAINT CLOCK OPERATIONS--
2980
2981  013502  012737  000107  001204          MOV     #71.,REPT
2982  013510  104446                  MRRD2:  MCLK1                   ;CLOCK MR REG
2983  013512  104420                          MRCK                    ;CHECK MR REG
2984  013514  033611                          33611                   ;TO EQUAL 33611
2985  013516  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2986  013520  104450                          MCLK0                   ;CLOCK MR REG
2987  013522  104420                          MRCK                    ;CHECK MR REG
2988  013524  023601                          23601                   ;TO EQUAL 23601
2989  013526  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2990  013530  005337  001204                  DEC     REPT            ;DONE YET
2991  013534  001365                          BNE     MRRD2           ;NO LOOP
2992
2993                                    ;READ SYNC"1"
2994
2995  013536  012777  000005  165360          MOV     #5,@RSMR
2996  013544  012777  000015  165352          MOV     #15,@RSMR
2997  013552  104420                          MRCK
2998  013554  133615                          133615
2999  013556  104000                          HLT
3000
3001                                    ;READ DATA
3002  013560  005037  001226          MRD3:   CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
3003  013564  012705  030114                  MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
3004  013570  162705  000002                  SUB     #2,R5
3005  013574  012737  000045  001206          MOV     #45,REPT1       ;SETUP COUNTER FOR 1ST SB BIT
3006  013602  012737  002200  001204          MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
3007                                                                  ;64 WORDS-18X64=1152
3008                                                                  ;1 CLOCK PER 1 BIT OF DATA
3009  013610  104444                  1$:     RBIT                    ;GET 1 DATA BIT
3010  013612  104440                          CLKR1                   ;CLOCK MR REG
3011  013614  104000                          HLT                     ;MR NOT CORRECT
3012
3013  013616  104442                          CLKR0                   ;CLOCK MR REG
3014  013620  104000                          HLT                     ;MR REG NOT CORRECT
3015
3016  013622  005337  001204                  DEC     REPT            ;DONE WITH COMPLETE TRANSFER
3017  013626  001370                          BNE     1$              ;NO
```

```
3018  013630  032737  000400  001170  2$:    BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
3019  013636  001030                         BNE     3$              ;YES
3020  013640  052737  000400  001170         BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
3021  013646  013737  001206  001174         MOV     REPT1,SAVEE     ;SAVE REPT1
3022  013654  004737  024430                 JSR     PC,GENCRC       ;GENERATE CRC WORD
3023                                                                 ;AND LEAVE IN LOC "WORK"
3024  013660  012702  030114                 MOV     #INBUF,R2
3025  013664  013737  001174  001206         MOV     SAVEE,REPT1     ;RESTORE REPT1
3026  013672  062702  000200                 ADD     #200,R2         ;STORE CRC WORD AT END OF
3027  013676  013712  001216                 MOV     WORK,@R2        ;INBUF TABLE
3028  013702  010205                         MOV     R2,R5
3029  013704  162705  000002                 SUB     #2,R5
3030  013710  012737  000022  001204         MOV     #18.,REPT       ;SETUP TO TRANSFER 1 WD
3031  013716  000734                         BR      1$              ;TRANSFER CRC WD
3032  013720  104446                 3$:     MCLK1                   ;CLOCK MR REG
3033  013722  104420                         MRCK                    ;CHECK MR REG
3034  013724  117611                         117611                  ;TO EQUAL
3035  013726  104000                         HLT                     ;117611
3036  013730  104450                         MCLKO                   ;CLOCK MR REG
3037  013732  104420                         MRCK                    ;CHECK MR
3038  013734  003601                         3601                    ;TO EQUAL
3039  013736  104000                         HLT                     ;3601
3040  013740  104446                         MCLK1                   ;CLOCK MR REG
3041  013742  104420                         MRCK                    ;CHECK MR
3042  013744  113611                         113611                  ;TO EQUAL
3043  013746  104000                         HLT                     ;113611
3044  013750  104450                         MCLKO                   ;CLOCK MR REG
3045  013752  104420                         MRCK                    ;CHECK MR
3046  013754  003601                         3601                    ;TO EQUAL
3047  013756  104000                         HLT                     ;3601
3048
3049                                 ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
3050                                 ;STEP INTO END OF SECTOR DEAD BAND
3051                                 ;EBL IS NOW ASSERTED
3052
3053  013760  012737  000020  001204 MRD4:   MOV     #20,REPT
3054  013766  104446                 1$:     MCLK1                   ;CLOCK MR REG
3055  013770  104420                         MRCK                    ;CHECK MR REG
3056  013772  113611                         113611                  ;TO EQUAL
3057  013774  104000                         HLT                     ;113611
3058  013776  104450                         MCLKO                   ;CLOCK MR REG
3059  014000  104420                         MRCK                    ;CHECK MR
3060  014002  003601                         3601                    ;REG TO
3061  014004  104000                         HLT                     ;EQUAL 3601
3062  014006  005337  001204                 DEC     REPT            ;DONE YET?
3063  014012  001365                         BNE     1$              ;NO
3064
3065                                 ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3066                                 ;SHOULD GET STROBE BUFFER
3067
3068  014014  104446                         MCLK1                   ;CLOCK MR REG
3069  014016  104420                         MRCK                    ;CHECK MR
3070  014020  117611                         117611                  ;REG TO
3071  014022  104000                         HLT                     ;EQUAL 117611
```

```
                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.
                                
014024  104450                  MRD5:   MCLKO                   ;CLOCK MR REG
014026  022777  004270  165044          CMP     #4270,@RSCS1    ;ANY ERRORS?
014034  001401                          BEQ     1$              ;NO
014036  104054                          HLT     !DA!DS!WC
014040  005777  165040          1$:     TST     @RSWC           ;DID WC GO TO 0
014044  001401                          BEQ     .+4             ;YES
014046  104010                          HLT     !WC             ;WC REG SHOULD=0
014050  022777  000001  165032          CMP     #1,@RSDA        ;DOES RSAD=1
014056  001401                          BEQ     .+4             ;YES
014060  104004                          HLT     !DA             ;NO RSDA SHOULD=1

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

014062  042737  000040  001170  MRD6:   BIC     #BIT5,ONCEE     ;CLEAR CLOCK TYPEOUT
014070  012701  030114                  MOV     #INBUF,GOOD     ;GET STARTING LOC OF EXPECTED DATA
014074  012700  030714                  MOV     #OUTBUF,BAD     ;GET STARTING LOC OF DATA "READ" FROM DISK
014100  012737  000012  001204          MOV     #12,REPT        ;SET UP ERROR COUNTER
014106  012705  000101                  MOV     #101,R5         ;COMPARE 1 SECTOR
014112  005305                  3$:     DEC     R5              ;DONE WITH SECTOR
014114  001445                          BEQ     2$              ;YES GET OUT
014116  022021                          CMP     (BAD)+,(GOOD)+  ;IS DATA CORRECT?
014120  001774                          BEQ     3$              ;YES
014122  032777  010000  164676          BIT     #BIT12,@SWR     ;TYPE ALL ERRORS?
014130  001003                          BNE     1$              ;YES
014132  005337  001204                  DEC     REPT            ;TYPED OUT 10 ERRORS YET?
014136  001434                          BEQ     2$              ;YES GET OUT
014140  024041                  1$:     CMP     -(BAD),-(GOOD)  ;GET ERROR
014142  010003                          MOV     BAD,R3          ;SAVE CONTENTS OF
014144  010104                          MOV     GOOD,R4         ;GOOD AND BAD
014146  011300                          MOV     (R3),BAD        ;GET DATA & PLEASE
014150  011401                          MOV     (R4),GOOD       ;IN GOOD AND BAD
014152  104000                          HLT                     ;TYPE OUT ERROR
014154  010300                          MOV     R3,BAD          ;PUT ADDRESS BACK
014156  010401                          MOV     R4,GOOD         ;INTO GOOD & BAD
014160  010037  001216                  MOV     BAD,WORK
014164  032777  020000  164634          BIT     #BIT13,@SWR
014172  001014                          BNE     4$
014174  104402  014200                  TYPE    ,.+2            ;.ASCIZ "BAD ADDRESS= "
014216  013746  001216                  MOV     WORK,-(6)       ;PUT WORK ON STACK
014222  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
014224  022021                  4$:     CMP     (BAD)+,(GOOD)+
014226  000731                          BR      3$
014230                          2$:                             ;DONE
```

```
                        ;*****************************************************************************
                        ;TEST 56                MAINTENANCE MODE DATA WRITE CHECK TEST
                        ;*****************************************************************************
014230  104400         TST56:  SCOPE

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

014232  104414         MRWCK:  CLRDK                   ;CLEAR DRIVE REGISTERS
014234  052737  000040  001170          BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
014242  042737  047716  001170          BIC     #47716,ONCEE    ;CLEAR ALL OTHER FLAG BITS
014250  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
014252  104420                          MRCK                    ;CHECK MR REG
014254  022701                          22701                   ;TO EQUAL 22701
014256  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES

014260  012737  000004  001144          MOV     #4,FLAG2        ;SET WC FLAG FOR CLKR1 ROUTINE

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

014266  012702  030114          MOV     #INBUF,R2       ;GET LOCATION OF INBUF
014272  005022                  CLR     (R2)+           ;CLEAR 1ST LOCATION
014274  012722  177777          MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
014300  005003                  CLR     R3              ;CLEAR WORK LOC TO GENERATE
014302  000261                  SEC                     ;A PATTERN OF FLOATING ONES
014304  006103          1$:     ROL     R3              ;GET PATTERN
014306  103402                  BCS     2$              ;DONE GET OUT
014310  010322                  MOV     R3,(R2)+        ;FILL BUFFER
014312  000774                  BR      1$              ;CONT
014314  012703  000056  2$:     MOV     #46.,R3         ;FILL REMAINING PORTION OF
014320  012704  146314          MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
014324  010422          3$:     MOV     R4,(R2)+        ;LOAD BUFFER
014326  005303                  DEC     R3              ;DONE YET
014330  001375                  BNE     3$              ;NO

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

014332  012777  030114  164546          MOV     #INBUF,@RSBA    ;LOAD BUS ADDR REG
014340  012777  177700  164536          MOV     #177700,@RSWC   ;LOAD WORD COUNT REG
014346  012777  000051  164524          MOV     #51,@RSCS1      ;LOAD WRITE CHECK COMMAND
014354  104454                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
014356  104220                  HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
014360  104456                  SPASS                   ;CLOCK MR REG SP = 1
```

```
3172
3173  014362  104430                    ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3174  014364  104420                            MRIND
3175  014366  022701                            MRCK                    ;CHECK MR REG TO EQUAL
3176  014370  104000                            22701                   ;22701
3177                                            HLT
3178
3179                                    ;STEP THRU RESYNC PERIOD
3180  014372  012737  001000  001204            MOV     #512.,REPT
3181  014400  052737  000040  001170            BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
3182  014406  104446            MRWCK1: MCLK1                           ;CLOCK MR REG
3183  014410  104420                    MRCK                            ;CHECK FOR
3184  014412  032711                    32711                           ;CORRECT DATA
3185  014414  104000                    HLT                             ;MR=BAD GOOD=CORRECT DATA
3186  014416  104450                    MCLK0                           ;CLOCK MR REG
3187  014420  104420                    MRCK                            ;CHECK FOR
3188  014422  022701                    22701                           ;CORRECT DATA
3189  014424  104000                    HLT                             ;ERROR WHILE CLOCKING THROUGH RESYNC
3190  014426  005337  001204            DEC     REPT                    ;FINISH LOOPING
3191  014432  001365                    BNE     MRWCK1                  ;THROUGH RESYNC PERIOD
3192
3193                                    ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3194                                    ;SP=0 EQUALS SECTOR PULSE
3195  014434  104446                    MCLK1                           ;CLOCK MR REG
3196  014436  104420                    MRCK                            ;MR SHOULD
3197  014440  032311                    32311                           ;EQUAL 32311
3198  014442  104000                    HLT                             ;MR=BAD GOOD=CORRECT ANS
3199  014444  104450                    MCLK0                           ;CLOCK MR REG
3200  014446  104420                    MRCK                            ;CHECK MR
3201  014450  022301                    22301                           ;TO EQUAL 22301
3202  014452  104000                    HLT                             ;MR=BAD GOOD=CORRECT ANS
3203
3204                                    ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--
3205
3206  014454  012737  000107  001204            MOV     #71.,REPT
3207  014462  104446            MRWCK2: MCLK1                           ;CLOCK MR REG
3208  014464  104420                    MRCK                            ;CHECK MR REG
3209  014466  033711                    33711                           ;TO EQUAL 33711
3210  014470  104000                    HLT                             ;MR=BAD GOOD=CORRECT ANS
3211  014472  104450                    MCLK0                           ;CLOCK MR REG
3212  014474  104420                    MRCK                            ;CHECK MR REG
3213  014476  023701                    23701                           ;TO EQUAL 23701
3214  014500  104000                    HLT                             ;MR=BAD GOOD=CORRECT ANS
3215  014502  005337  001204            DEC     REPT                    ;DONE YET
3216  014506  001365                    BNE     MRWCK2                  ;NO LOOP
```

```
3217                                        ;READ SYNC"1"
3218
3219  014510  012777  000005  164406          MOV     #5,@RSMR
3220  014516  012777  000015  164400          MOV     #15,@RSMR
3221  014524  104420                          MRCK
3222  014526  133715                          133715
3223  014530  104000                          HLT
3224
3225                                        ;READ DATA
3226  014532  005037  001226          MRWCK3: CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
3227  014536  012705  030114                  MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
3228  014542  162705  000002                  SUB     #2,R5
3229  014546  012737  000045  001206          MOV     #45,REPT1       ;SETUP COUNTER FOR 1ST SB BIT
3230  014554  012737  002200  001204          MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
3231                                                                  ;64 WORDS-18X64=1152
3232                                                                  ;1 CLOCK PER 1 BIT OF DATA
3233  014562  104444                  1$:     RBIT                    ;GET 1 DATA BITS
3234  014564  104440                          CLKR1                   ;CLOCK MR REG
3235  014566  104000                          HLT                     ;MR NOT CORRECT
3236
3237  014570  104442                          CLKR0                   ;CLOCK MR REG
3238  014572  104000                          HLT                     ;MR REG NOT CORRECT
3239
3240  014574  005337  001204                  DEC     REPT            ;DONE WITH COMPLETE TRANSFER
3241  014600  001370                          BNE     1$              ;NO
3242  014602  032737  000400  001170  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
3243  014610  001030                          BNE     3$              ;YES
3244  014612  052737  000400  001170          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
3245  014620  013737  001206  001174          MOV     REPT1,SAVEE     ;SAVE REPT1
3246  014626  004737  024430                  JSR     PC,GENCRC       ;GENERATE CRC WORD
3247                                                                  ;AND LEAVE IN LOC "WORK"
3248  014632  012702  030114                  MOV     #INBUF,R2       ;
3249  014636  013737  001174  001206          MOV     SAVEE,REPT1     ;RESTORE REPT1
3250  014644  062702  000200                  ADD     #200,R2         ;STORE CRC WORD AT END OF
3251  014650  013712  001216                  MOV     WORK,@R2        ;INBUF TABLE
3252  014654  010205                          MOV     R2,R5
3253  014656  162705  000002                  SUB     #2,R5
3254  014662  012737  000022  001204          MOV     #18.,REPT       ;SETUP TO TRANSFER 1 WD
3255  014670  000734                          BR      1$              ;TRANSFER CRC WD
```

```
3256  014672  104446                    3$:     MCLK1                   ;CLOCK MR REG
3257  014674  104420                            MRCK                    ;CHECK MR REG
3258  014676  117711                            117711                  ;TO EQUAL
3259  014700  104000                            HLT                     ;117711
3260  014702  104450                            MCLKO                   ;CLOCK MR REG
3261  014704  104420                            MRCK                    ;CHECK MR
3262  014706  003701                            3701                    ;TO EQUAL
3263  014710  104000                            HLT                     ;3701
3264  014712  104446                            MCLK1                   ;CLOCK MR REG
3265  014714  104420                            MRCK                    ;CHECK MR
3266  014716  113711                            113711                  ;TO EQUAL
3267  014720  104000                            HLT                     ;113711
3268  014722  104450                            MCLKO                   ;CLOCK MR REG
3269  014724  104420                            MRCK                    ;CHECK MR
3270  014726  003701                            3701                    ;TO EQUAL
3271  014730  104000                            HLT                     ;3701
3272
3273                                  ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
3274                                  ;STEP INTO END OF SECTOR DEAD BAND
3275                                  ;FBL IS NOW ASSERTED
3276
3277  014732  012737  000020  001204  MRWCK4: MOV     #20,REPT
3278  014740  104446                  1$:     MCLK1                   ;CLOCK MR REG
3279  014742  104420                          MRCK                    ;CHECK MR REG
3280  014744  113711                          113711                  ;TO EQUAL
3281  014746  104000                          HLT                     ;113711
3282  014750  104450                          MCLKO                   ;CLOCK MR REG
3283  014752  104420                          MRCK                    ;CHECK MR
3284  014754  003701                          3701                    ;REG TO
3285  014756  104000                          HLT                     ;EQUAL 3701
3286  014760  005337  001204                  DEC     REPT            ;DONE YET?
3287  014764  001365                          BNE     1$              ;NO
3288
3289                                  ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3290                                  ;SHOULD GET STROBE BUFFER
3291
3292  014766  104446                          MCLK1                   ;CLOCK MR REG
3293  014770  104420                          MRCK                    ;CHECK MR
3294  014772  117711                          117711                  ;REG TO
3295  014774  104000                          HLT                     ;EQUAL 117711
3296
3297                                  ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3298                                  ;SHOULD COMPLETE TRANSFER.
3299
3300  014776  104450                  MRWCK5: MCLKO                   ;CLOCK MR REG
3301  015000  022777  004250  164072          CMP     #4250,@RSCS1    ;ANY ERRORS?
3302  015006  001401                          BEQ     1$              ;NO
3303  015010  104054                          HLT     !DA!DS!WC
3304  015012  005777  164066          1$:     TST     @RSWC           ;DID WC GO TO 0
3305  015016  001401                          BEQ     .+4             ;YES
3306  015020  104010                          HLT     !WC             ;WC REG SHOULD=0
3307  015022  022777  000001  164060          CMP     #1,@RSDA        ;DOES RSDA=1
3308  015030  001401                          BEQ     .+4             ;YES
3309  015032  104004                          HLT     !DA             ;RSDA SHOULD=1
```

```
                                ;*****************************************************************************
                                ;TEST 57                    MAINTENANCE MODE CRC TEST 1 (NO DCK ERRORS)
                                ;*****************************************************************************
015034  104400                  TST57:  SCOPE

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

015036  012737  000040  001144          MOV     #40,FLAG2       ;CLEAR TST FLAG
015044  104414  MRCRC:  CLRDK           ;CLEAR DRIVE REGISTERS
015046  052737  000040  001170          BIS     #BIT5,ONCEE     ;TYPE CLOCK COUNT IF ERROR OCCURS
015054  042737  047716  001170          BIC     #47716,ONCEE    ;CLEAR ALL OTHER FLAG BITS
015062  104430                  MRIND   ;SEND INDEX PULSE TO MR REG
015064  104420                  MRCK    ;CHECK MR REG
015066  022701                  22701   ;TO EQUAL 22701
015070  104424                  MRINT   ;INIT MAINT MODE (CLEAR MRSP)
                                        ;BY SENDING 2 CLOCK PULSES
015072  032737  000020  001144          BIT     #BIT4,FLAG2     ;FIRST TIME THROUGH
015100  001023                  BNE     3$              ;NO
015102  012737  000001  001174          MOV     #1,SAVEE        ;LOAD 1ST CRC WORD

                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 1 SECTOR
                                ;CREATE BUFFER WITH 72 WORDS OF 16 BITS WHICH EQUALS THE NO. OF BITS IN 64 18 BITS WORDS
                                ;DATA BUFFER CONTAINS 6 WORDS OF ZEROS
                                ;                    A WORD OF 236
                                ;                    A WORD OF 140000
                                ;                    64 WORDS OF ZEROS
                                ;IN THIS TEST, ALL 18 BITS OF THE RS03 DATA WORD MUST BE
                                ;MANIPULATED.  HENCE A TABLE CONTAINING 1152 BITS (64X18) IS
                                ;REQUIRED INSTEAD OF A TABLE CONTAINING 64 WORDS.
015110  012702  030114                  MOV     #INBUF,R2       ;GET LOCATION OF INBUF
015114  012703  000006                  MOV     #6,R3           ;SETUP COUNTER
015120  005022          1$:     CLR     (R2)+           ;TO CLEAR THE
015122  005303                  DEC     R3              ;FIRST 6
015124  001375                  BNE     1$              ;WORDS
015126  012722  000236                  MOV     #236,(R2)+      ;LOAD A 236
015132  012722  140000                  MOV     #140000,(R2)+   ;LOAD A 140000
015136  012703  000100                  MOV     #64.,R3         ;SETUP COUNTER
015142  005022          2$:     CLR     (R2)+           ;TO CLEAR THE
015144  005303                  DEC     R3              ;REMAINING WORDS
015146  001375                  BNE     2$              ;FOR
```

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

015150  012777  030714  163730  3$:     MOV     #OUTBUF,@RSBA   ;LOAD BUS ADDR REG
015156  012777  177700  163720          MOV     #177700,@RSWC   ;LOAD WORD COUNT REG
015164  012777  000071  163706          MOV     #71,@RSCS1      ;LOAD READ COMMAND
015172  012702  000200                  MOV     #200,R2
015176  012703  030714                  MOV     #OUTBUF,R3
015202  052737  000020  001144          BIS     #BIT4,FLAG2     ;NO SET FLAG FOR 1ST TIME THROUGH TEST
015210  005023                  4$:     CLR     (R3)+
015212  005302                          DEC     R2
015214  001375                          BNE     4$
015216  104454                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
015220  104220                          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
015222  104456                          SPASS                   ;CLOCK MR REG SP = 1

                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
015224  104430                          MRIND
015226  104420                          MRCK                    ;CHECK MR REG TO EQUAL
015230  022601                          22601                   ;22601 FOR A
015232  104000                          HLT                     ;READ COMD

                                        ;STEP THRU RESYNC PERIOD

015234  012737  001000  001204          MOV     #512.,REPT
015242  052737  000040  001170          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
015250  104446                  MRCRC1: MCLK1                   ;CLOCK MR REG
015252  104420                          MRCK                    ;CHECK FOR
015254  032611                          32611                   ;CORRECT DATA
015256  104000                          HLT                     ;MR=BAD GOOD=CORRECT DATA
015260  104450                          MCLK0                   ;CLOCK MR REG
015262  104420                          MRCK                    ;CHECK FOR
015264  022601                          22601                   ;CORRECT DATA
015266  104000                          HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
015270  005337  001204                  DEC     REPT            ;FINISH LOOPING
015274  001365                          BNE     MRCRC1          ;THROUGH RESYNC PERIOD

                                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                ;SP=0 EQUALS SECTOR PULSE
015276  104446                          MCLK1                   ;CLOCK MR REG
015300  104420                          MRCK                    ;MR SHOULD
015302  032211                          32211                   ;EQUAL 32211
015304  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
015306  104450                          MCLK0                   ;CLOCK MR REG
015310  104420                          MRCK                    ;CHECK MR
015312  022201                          22201                   ;TO EQUAL 22201
015314  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                        ;PERFORM 71 MAINT CLOCK OPERATIONS--

015316  012737  000107  001204          MOV     #71.,REPT
015324  104446                  MRCRC2: MCLK1                   ;CLOCK MR REG
015326  104420                          MRCK                    ;CHECK MR REG
015330  033611                          33611                   ;TO EQUAL 33611
015332  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
015334  104450                          MCLK0                   ;CLOCK MR REG
015336  104420                          MRCK                    ;CHECK MR REG
015340  023601                          23601                   ;TO EQUAL 23601
015342  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
015344  005337  001204                  DEC     REPT            ;DONE YET
015350  001365                          BNE     MRCRC2          ;NO LOOP

                                        ;READ SYNC"1"
015352  012777  000005  163544          MOV     #5,@RSMR
015360  012777  000015  163536          MOV     #15,@RSMR
015366  104420                          MRCK
015370  133615                          133615
015372  104000                          HLT

                                        ;READ DATA
015374  005037  001226          MRCRC3: CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
015400  012705  030114                  MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
015404  162705  000002                  SUB     #2,R5
015410  012737  000045  001206          MOV     #45,REPT1       ;SETUP COUNTER FOR 1ST SB BIT
015416  012737  002200  001204          MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;64 WORDS-18X64=1152
                                                                ;1 CLOCK PER 1 BIT OF DATA
015424  104444                  1$:     RBIT                    ;GET 1 DATA BITS
015426  104440                          CLKR1                   ;CLOCK MR REG
015430  104000                          HLT                     ;MR NOT CORRECT

015432  104442                          CLKR0                   ;CLOCK MR REG
015434  104000                          HLT                     ;MR REG NOT CORRECT

015436  005337  001204                  DEC     REPT            ;DONE WITH COMPLETE TRANSFER
015442  001370                          BNE     1$              ;NO
```

```
 3443  015444  032737  000400  001170  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
 3444  015452  001020                          BNE     3$              ;YES
 3445  015454  052737  000400  001170          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
 3446  015462  012702  030114                  MOV     #INBUF,R2       ;MOVE CRC
 3447  015466  062702  000220                  ADD     #220,R2         ;WORD TO END OF
 3448  015472  013712  001174          4$:     MOV     SAVEE,@R2       ;INBUF TABLE
 3449  015476  010205                  5$:     MOV     R2,R5           ;GET CRC WORD
 3450  015500  162705  000002                  SUB     #2,R5
 3451  015504  012737  000022  001204          MOV     #18.,REPT       ;SETUP TO TRANSFER 1 WD
 3452  015512  000744                          BR      1$              ;TRANSFER CRC WD
 3453  015514  104446                  3$:     MCLK1                   ;CLOCK MR REG
 3454  015516  104420                          MRCK                    ;CHECK MR REG
 3455  015520  117611                          117611                  ;TO EQUAL
 3456  015522  104000                          HLT                     ;117611
 3457  015524  104450                          MCLKO                   ;CLOCK MR REG
 3458  015526  104420                          MRCK                    ;CHECK MR
 3459  015530  003601                          3601                    ;TO EQUAL
 3460  015532  104000                          HLT                     ;3601
 3461  015534  104446                          MCLK1                   ;CLOCK MR REG
 3462  015536  104420                          MRCK                    ;CHECK MR
 3463  015540  113611                          113611                  ;TO EQUAL
 3464  015542  104000                          HLT                     ;113611
 3465  015544  104450                          MCLKO                   ;CLOCK MR REG
 3466  015546  104420                          MRCK                    ;CHECK MR
 3467  015550  003601                          3601                    ;TO EQUAL
 3468  015552  104000                          HLT                     ;3601
```

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

015554  012737  000020  001204  MRCRC4: MOV     #20,REPT

015562  104446                  1$:     MCLK1                   ;CLOCK MR REG
015564  104420                          MRCK                    ;CHECK MR REG
015566  113611                          113611                  ;TO EQUAL
015570  104000                          HLT                     ;113611
015572  104450                          MCLKO                   ;CLOCK MR REG
015574  104420                          MRCK                    ;CHECK MR
015576  003601                          3601                    ;REG TO
015600  104000                          HLT                     ;EQUAL 3601
015602  005337  001204                  DEC     REPT            ;DONE YET?
015606  001365                          BNE     1$              ;NO

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD GET STROBE BUFFER

015610  104446                          MCLK1                   ;CLOCK MR REG
015612  104420                          MRCK                    ;CHECK MR
015614  117611                          117611                  ;REG TO
015616  104000                          HLT                     ;EQUAL 117611

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.

015620  104450                  MRCRC5: MCLKO                   ;CLOCK MR REG
015622  022777  004270  163250          CMP     #4270,@RSCS1    ;ANY ERRORS?
015630  001401                          BEQ     1$              ;NO
015632  104054                          HLT     !DA!DS!WC
015634  005777  163244          1$:     TST     @RSWC           ;DID WC GO TO 0
015640  001401                          BEQ     .+4             ;YES
015642  104010                          HLT     !WC             ;WC REG SHOULD=0
015644  006137  001174                  ROL     SAVEE           ;GET NEXT CRC WORD
015650  103404                          BCS     2$              ;DONE - BRANCH
015652  004737  027110                  JSR     PC,MDATA        ;SHIFT DATA PATTERN
015656  000137  015044                  JMP     MRCRC           ;RESTART TEST WITH NEW DATA PATTERN
015662                          2$:     ;DONE
```

```
                        ;*************************************************************************
                        ;TEST 60                MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)
                        ;*************************************************************************
015662  104400          TST60:  SCOPE

                        ;MODULE TESTED M7753
                        ;THIS TEST IS SIMILAR TO CRC TEST 1 EXCEPT THAT THE DATA
                        ;PATTERN HAS BEEN MODIFIED TO LEAVE A SINGLE BIT SET IN THE
                        ;CRC REGISTER AFTER BOTH DATA AND CRC WORDS HAVE BEEN "READ".
                        ;THIS CAUSES A DCK ERROR.  THE READ SEQUENCE IS REPEATED 16
                        ;TIMES TO TEST THAT EACH BIT IN THE CRC REGISTER CAN CAUSE A
                        ;DCK ERROR.
015664  012737  000040  001144          MOV     #40,FLAG2       ;CLEAR TST FLAG
015672  104414  MRDCK:   CLRDK                                  ;CLEAR DRIVE REGISTERS
015674  052737  000040  001170          BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
015702  042737  047716  001170          BIC     #47716,ONCEE    ;CLEAR ALL OTHER FLAG BITS
015710  104430          MRIND                                   ;SEND INDEX PULSE TO MR REG
015712  104420          MRCK                                    ;CHECK MR REG
015714  022701          22701                                   ;TO EQUAL 22701
015716  104424          MRINT                                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
015720  032737  000020  001144          BIT     #BIT4,FLAG2     ;FIRST TIME THROUGH
015726  001023          BNE     3$      ;NO
015730  012737  000001  001174          MOV     #1,SAVEE        ;LOAD 1ST CRC WORD

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

015736  012702  030114          MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
015742  012703  000007          MOV     #7,R3           ;SETUP COUNTER
015746  005022  1$:     CLR     (R2)+                   ;TO CLEAR THE
015750  005303          DEC     R3                      ;FIRST 7
015752  001375          BNE     1$                      ;WORDS
015754  012722  000023          MOV     #23,(R2)+       ;LOAD A 23
015760  012722  154000          MOV     #154000,(R2)+   ;LOAD A 154000
015764  012703  000077          MOV     #63.,R3         ;SETUP COUNTER
015770  005022  2$:     CLR     (R2)+                   ;TO CLEAR THE
015772  005303          DEC     R3                      ;REMAINING WORDS
015774  001375          BNE     2$                      ;FOR THAT SECTOR

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

015776  012777  030714  163102  3$:     MOV     #OUTBUF,@RSBA   ;LOAD BUS ADDR REG
016004  012777  177700  163072          MOV     #177700,@RSWC   ;LOAD WORD COUNT REG
016012  012777  000071  163060          MOV     #71,@RSCS1      ;LOAD READ COMMAND
016020  012702  000200          MOV     #200,R2
016024  012703  030714          MOV     #OUTBUF,R3
016030  052737  000020  001144          BIS     #BIT4,FLAG2     ;NO SET FLAG FOR 1ST TIME THROUGH TEST
016036  005023  4$:     CLR     (R3)+
016040  005302          DEC     R2
016042  001375          BNE     4$
```

```
3563  016044  104454                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
3564                                                                  ;TO CLEAR OUT COUNTERS AND REGISTERS
3565                                                                  ;THAT OTHERWISE COULD NOT BE CLEARED.
3566  016046  104220                          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
3567  016050  104456                          SPASS                   ;CLOCK MR REG SP = 1
3568                                  ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3569  016052  104430                          MRIND
3570  016054  104420                          MRCK                    ;CHECK MR REG TO EQUAL
3571  016056  022601                          22601                   ;22601
3572  016060  104000                          HLT
3573
3574                                  ;STEP THRU RESYNC PERIOD
3575
3576  016062  012737  001000  001204          MOV     #512.,REPT
3577  016070  052737  000040  001170          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT
3578  016076  104446                  MRDCK1: MCLK1                   ;CLOCK MR REG
3579  016100  104420                          MRCK                    ;CHECK FOR
3580  016102  032611                          32611                   ;CORRECT DATA
3581  016104  104000                          HLT                     ;MR=BAD GOOD=CORRECT DATA
3582  016106  104450                          MCLK0                   ;CLOCK MR REG
3583  016110  104420                          MRCK                    ;CHECK FOR
3584  016112  022601                          22601                   ;CORRECT DATA
3585  016114  104000                          HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
3586  016116  005337  001204                 DEC     REPT            ;FINISH LOOPING
3587  016122  001365                          BNE     MRDCK1          ;THROUGH RESYNC PERIOD
3588
3589                                  ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3590                                  ;SP=0 EQUALS SECTOR PULSE
3591  016124  104446                          MCLK1                   ;CLOCK MR REG
3592  016126  104420                          MRCK                    ;MR SHOULD
3593  016130  032211                          32211                   ;EQUAL 32211
3594  016132  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
3595  016134  104450                          MCLK0                   ;CLOCK MR REG
3596  016136  104420                          MRCK                    ;CHECK MR
3597  016140  022201                          22201                   ;TO EQUAL 22201
3598  016142  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                        ;PERFORM 71 MAINT CLOCK OPERATIONS--

016144  012737  000107  001204          MOV     #71.,REPT
016152  104446                  MRDCK2: MCLK1                   ;CLOCK MR REG
016154  104420                          MRCK                    ;CHECK MR REG
016156  033611                          33611                   ;TO EQUAL 33611
016160  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
016162  104450                          MCLKO                   ;CLOCK MR REG
016164  104420                          MRCK                    ;CHECK MR REG
016166  023601                          23601                   ;TO EQUAL 23601
016170  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
016172  005337  001204                  DEC     REPT            ;DONE YET
016176  001365                          BNE     MRDCK2          ;NO LOOP

                                        ;READ SYNC"1"
016200  012777  000005  162716          MOV     #5,@RSMR
016206  012777  000015  162710          MOV     #15,@RSMR
016214  104420                          MRCK
016216  133615                          133615
016220  104000                          HLT

                                        ;READ DATA
016222  005037  001226          MRDCK3: CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
016226  012705  030114                  MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
016232  162705  000002                  SUB     #2,R5
016236  012737  000045  001206          MOV     #45,REPT1       ;SETUP COUNTER FOR 1ST SB BIT
016244  012737  002200  001204          MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;64 WORDS-18X64=1152
                                                                ;1 CLOCK PER 1 BIT OF DATA
016252  104444                  1$:     RBIT                    ;GET 1 DATA BITS
016254  104440                          CLKR1                   ;CLOCK MR REG
016256  104000                          HLT                     ;MR NOT CORRECT

016260  104442                          CLKRO                   ;CLOCK MR REG
016262  104000                          HLT                     ;MR REG NOT CORRECT

016264  005337  001204                  DEC     REPT            ;DONE WITH COMPLETE TRANSFER
016270  001370                          BNE     1$              ;NO
```

```
3637  016272  032737  000400  001170  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
3638  016300  001020                          BNE     3$              ;YES
3639  016302  052737  000400  001170          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
3640  016310  012702  030114                  MOV     #INBUF,R2       ;MOVE CRC
3641  016314  062702  000220                  ADD     #220,R2         ;WORD TO END OF
3642  016320  012712  000000          4$:     MOV     #0,@R2          ;INBUF TABLE
3643  016324  010205                  5$:     MOV     R2,R5           ;GET CRC WORD
3644  016326  162705  000002                  SUB     #2,R5
3645  016332  012737  000022  001204          MOV     #18.,REPT       ;SETUP TO TRANSFER 1 WD
3646  016340  000744                          BR      1$              ;TRANSFER CRC WD
3647  016342  104446                  3$:     MCLK1                   ;CLOCK MR REG
3648  016344  104420                          MRCK                    ;CHECK MR REG
3649  016346  117611                          117611                  ;TO EQUAL
3650  016350  104000                          HLT                     ;117611
3651  016352  104450                          MCLK0                   ;CLOCK MR REG
3652  016354  104420                          MRCK                    ;CHECK MR
3653  016356  003601                          3601                    ;TO EQUAL
3654  016360  104000                          HLT                     ;3601
3655  016362  104446                          MCLK1                   ;CLOCK MR REG
3656  016364  104420                          MRCK                    ;CHECK MR
3657  016366  113611                          113611                  ;TO EQUAL
3658  016370  104000                          HLT                     ;113611
3659  016372  104450                          MCLK0                   ;CLOCK MR REG
3660  016374  104420                          MRCK                    ;CHECK MR
3661  016376  003601                          3601                    ;TO EQUAL
3662  016400  104000                          HLT                     ;3601
```

```
                                        ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
                                        ;STEP INTO END OF SECTOR DEAD BAND
                                        ;EBL IS NOW ASSERTED
                
016402  012737  000020  001204  MRDCK4: MOV     #20,REPT
                
016410  104446                  1$:     MCLK1                   ;CLOCK MR REG
016412  104420                          MRCK                    ;CHECK MR REG
016414  113611                          113611                  ;TO EQUAL
016416  104000                          HLT                     ;113611
016420  104450                          MCLK0                   ;CLOCK MR REG
016422  104420                          MRCK                    ;CHECK MR
016424  003601                          3601                    ;REG TO
016426  104000                          HLT                     ;EQUAL 3601
016430  005337  001204                  DEC     REPT            ;DONE YET?
016434  001365                          BNE     1$              ;NO
                
                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD GET STROBE BUFFER
                
016436  104446                          MCLK1                   ;CLOCK MR REG
016440  104420                          MRCK                    ;CHECK MR
016442  117611                          117611                  ;REG TO
016444  104000                          HLT                     ;EQUAL 117611
                
                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.
                
016446  104450                  MRDCK5: MCLK0                   ;CLOCK MR REG
016450  022777  144270  162422          CMP     #144270,@RSCS1  ;ANY ERRORS?
016456  001401                          BEQ     1$              ;NO
016460  104054                          HLT     !DA!DS!WC
016462  005777  162416          1$:     TST     @RSWC           ;DID WC GO TO 0
016466  001401                          BEQ     .+4             ;YES
016470  104010                          HLT     !WC             ;WC REG SHOULD=0
016472  022777  100000  162414          CMP     #100000,@RSER   ;DID DCK SET?
016500  001417                          BEQ     3$              ;YES
016502  104050                          HLT     !DS!WC
016504  104402  016510                  TYPE    .+2             ;.ASCIZ <15><12>"DCK DID NOT SET "
016534  004737  023006                  JSR     PC,CRCTYP       ;GET IC THAT FAILED AND TYPE IT
016540  000241                  3$:     CLC
016542  006137  001174                  ROL     SAVEE           ;GET NEXT CRC WORD
016546  103404                          BCS     2$              ;DONE - BRANCH
016550  004737  027110                  JSR     PC,MDATA        ;SHIFT DATA PATTERN
016554  000137  015672                  JMP     MRDCK           ;RESTART TEST WITH NEW DATA PATTERN
016560                          2$:                             ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 61                    IGNORE FUNCTION TEST
                                ;*****************************************************************************
016560  104400                  TST61:  SCOPE

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

016562  104414          MRIFT:  CLRDK                   ;CLEAR ALL REGISTERS
016564  104416                  MRDMD                   ;PUT DRIVE INTO MAINT MODE
016566  104420                  MRCK                    ;CHECK MR REG
016570  022701                  22701                   ;TO EQUAL 22701
016572  104424                  MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
016574  012777  177777  162312  MOV     #-1,@RSER       ;SET ERRORS
016602  013777  001164  162306  MOV     UNITSV,@RSAS    ;CLEAR ATA BIT IN RSAS
                                                        ;AND ERROR BITS IN RSCS1
016610  012777  030714  162270  MOV     #OUTBUF,@RSBA   ;LOAD RSBA
016616  012777  177777  162260  MOV     #-1,@RSWC       ;LOAD RSWC
016624  012777  000071  162246  MOV     #71,@RSCS1      ;LOAD READ FUNCTION
016632  032777  000001  162240  BIT     #BIT0,@RSCS1    ;IS "GO" BIT ZERO?
016640  001401                  BEQ     1$              ;YES
016642  104140                  HLT     !DS!AS          ;"GO" BIT IN RSCS1 SHOULD NOT
                                                        ;LOAD IF ERRORS ARE PRESENT IN THE DRIVE
016644  012737  177777  001216  1$:     MOV     #177777,WORK    ;SETUP TIMEOUT FOR MXF ERROR
016652  005337  001216          5$:     DEC     WORK
016656  000240                  NOP
016660  000240                  NOP
016662  001373                  BNE     5$
016664  017700  162212          MOV     @RSCS2,BAD      ;CHECK RSCS2 FOR MXF
016670  012701  001100          MOV     #1100,GOOD      ;GET CORRECT ANS
016674  053701  001162          BIS     UNNUM,GOOD      ;FOR RSCS2
016700  020001                  CMP     BAD,GOOD        ;IS RSCS2 CORRECT
016702  001401                  BEQ     2$              ;YES
016704  104000                  HLT                     ;BAD=RSCS2 GOOD=CORRECT ANS
                                                        ;MXF SHOULD BE SET IN RSCS2
                                                        ;FOR A READ WAS ISSUED
                                                        ;WITH ERROR BITS SET IN RSER.
016706  022777  144270  162164  2$:     CMP     #144270,@RSCS1  ;IS RSCS1 CORRECT?
016714  001401                  BEQ     3$              ;YES
016716  104042                  HLT     !DS!ER          ;SC AND TRE SHOULD BE SET FOR
                                                        ;MXF SHOULD BE SET IN RSCS2
```

```
3753  016720  022777  177777  162156  3$:   CMP   #-1,@RSWC      ;DID RSWC CHANGE?
3754  016726  001401                  BEQ   .+4                  ;NO
3755  016730  104010                  HLT   !WC                  ;WC SHOULD NOT HAVE CHANGED
3756                                                             ;FOR READ SHOULD HAVE NEVER
3757                                                             ;BEEN EXECUTED
3758  016732  022777  030714  162146        CMP   #OUTBUF,@RSBA  ;DID RSBA MOVE
3759  016740  001405                  BEQ   4$                   ;NO
3760  016742  017700  162140          MOV   @RSBA,BAD            ;GET RSBA
3761  016746  012701  030714          MOV   #OUTBUF,GOOD         ;GET CORRECT ANS
3762  016752  104000                  HLT                        ;RSBA=BAD GOOD=CORRECT ANS
3763                                                             ;RSBA MOVED WHEN THE READ
3764                                                             ;COND WAS LOADED WITH ERROR
3765                                                             ;BITS SET IN RSER, READ COMD SHOULD
3766                                                             ;NEVER HAVE BEEN EXECUTED AND RSBA
3767                                                             ;SHOULD NOT HAVE MOVED.
3768  016754  104414          4$:   CLRDK                        ;CLEAR ALL REGISTERS
3769  016756  017700  162120          MOV   @RSCS2,BAD           ;GET RSCS2
3770  016762  012701  000100          MOV   #100,GOOD            ;SETUP CORRECT
3771  016766  053701  001162          BIS   UNNUM,GOOD           ;ANS
3772  016772  020100                  CMP   GOOD,BAD             ;IS CS2 CORRECT?
3773  016774  001401                  BEQ   .+4                  ;YES
3774  016776  104000                  HLT                        ;MXF SHOULD HAVE CLEARED IN
3775                                                             ;RSCS2 WITH THE "CLR" BIT BEING
3776                                                             ;SET IN RSCS2.
```

```
                                ;*****************************************************************************
                                ;TEST 62                    INVALID ADDRESS ERROR (IAE) TEST
                                ;*****************************************************************************
017000  104400                  TST62:  SCOPE

                                ;MODULE TESTED M7754, M7770
                                ;FLOAT A 1 THROUGH THE FOUR SPARE ADDRESS BITS IN THE DISK
                                ;ADDRESS REGISTER (RSDA).  THIS SHOULD CAUSE "IAE" TO SET IN
                                ;THE ERROR REGISTER (RSER) WHEN A READ FUNCTION IS LOADED INTO
                                ;RSCS1 WHICH IN TURN SHOULD CAUSE ATTENTION TO SET IN THE
                                ;DRIVE STATUS REGISTER (RSDS) AND "TRE" AND "SC" TO SET IN THE
                                ;CONTROL REGISTER (RSCS1).
017002  042737  000040  001170          BIC     #BIT5,ONCEE     ;CLEAR CLK CNT FLAG
017010  012702  004000                  MOV     #4000,R2        ;LOAD R2 WITH INVALID ADDR
017014  012737  017022  001010          MOV     #4$,LAD         ;LOOP HERE ON ERROR
017022  104416                  4$:     MRDMD                   ;PUT DRIVE IN MAINT MODE
017024  104420                          MRCK                    ;CHECK MAINT REG
017026  022701                          22701
017030  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
017032  032737  000004  001170          BIT     #BIT2,ONCEE     ;LOOPING ON ERRORS>
017040  001002                          BNE     1$              ;YES
017042  006102                          ROL     R2              ;GET INVALID ADDRESS
017044  103454                          BCS     IADONE          ;DONE FLOATING A ONE YET?
017046  010277  162036          1$:     MOV     R2,@RSDA        ;LOAD RSDA WITH INVALID ADDRESS
017052  012777  000071  162020          MOV     #71,@RSCS1      ;DO A READ TO INVALID ADDR
017060  022777  002000  162026          CMP     #2000,@RSER     ;IS RSER CORRECT?
017066  001404                          BEQ     2$              ;YES
017070  052737  000004  001170          BIS     #BIT2,ONCEE     ;SET ERROR BIT
017076  104044                          HLT     !DS!DA          ;RSER SHOULD=2000 FOR
                                                                ;A READ COMMAND WAS GIVEN
                                                                ;TO AN ILLEGAL ADDRESS
017100  042737  000004  001170  2$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
017106  022777  150600  161776          CMP     #150600,@RSDS   ;DID IAE SET?
017114  001404                          BEQ     3$              ;YES
017116  052737  000004  001170          BIS     #BIT2,ONCEE     ;SET ERROR BIT
017124  104044                          HLT     !DS!DA          ;RSDS SHOULD=150600 FOR
                                                                ;IAE SHOULD BE SET IN RSER
017126  042737  000004  001170  3$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
017134  022777  144270  161736          CMP     #144270,@RSCS1  ;DIS SC + TRE SET?
017142  001404                          BEQ     5$              ;YES
017144  052737  000004  001170          BIS     #BIT2,ONCEE     ;SET ERROR BIT
017152  104044                          HLT     !DA!DS          ;SC + TRE SHOULD BE SET IN RSCS1
                                                                ;FOR IAE SHOULD BE SET IN RSER
017154  042737  000004  001170  5$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR BIT
017162  104414                          CLRDK                   ;CLEAR ALL ERRORS
017164  005777  161724                  TST     @RSER           ;DID IAE CLEAR?
017170  001401                          BEQ     .+4             ;YES
017172  104040                          HLT     !DS             ;IAE DID NOT CLEAR
017174  000712                          BR      4$              ;CONTINUE
017176                          IADONE:                         ;DONE
```

```
                                ;*******************************************************************************
                                ;TEST 63                    OPERATION INCOMPLETE ERROR TEST
                                ;*******************************************************************************
017176  104400                  TST63:  SCOPE

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

017200  104414                  MROPI:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
017202  013777  030714  161676          MOV     @#OUTBUF,@RSBA  ;SETUP RSBA
017210  012777  177777  161666          MOV     #-1,@RSWC       ;SETUP RSWC

017216  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
017220  104420                          MRCK                    ;CHECK MAINT REG
017222  022701                          22701                   ;TO EQUAL 22701
017224  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)

017226  012777  000071  161644          MOV     #71,@RSCS1      ;LOAD A READ COMMAND

017234  104430                          MRIND                   ;ISSUE THREE INDEX
017236  104430                          MRIND                   ;PULSES TO
017240  104430                          MRIND                   ;CAUSE OPI

                                ;NOW CHECK FOR CORRECT ERRORS IN RSER AND RSDS
017242  017700  161646                  MOV     @RSER,BAD       ;GET RSER
017246  012701  020000                  MOV     #20000,GOOD     ;GET CORRECT ANS
017252  020100                          CMP     GOOD,BAD        ;DID OPI SET IN RSER?
017254  001434                          BEQ     1$              ;YES
017256  104402  017262                  TYPE    ,.+2            ;.ASCIZ <15><12>"OPI IN RSER SHOULD SET-3 INDEX PULSES W
017344  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS

017346  022777  150600  161536  1$:     CMP     #150600,@RSDS   ;DID CORRECT ERRORS SET?
017354  001401                          BEQ     2$              ;YES
017356  104040                          HLT     !DS             ;RSDS SHOULD=150600 BECAUSE
                                                                ;OF OPI ERROR IN RSER
017360  022777  144270  161512  2$:     CMP     #144270,@RSCS1  ;DID SC AND TRE SET IN RSCS1?
017366  001401                          BEQ     MROPIA          ;YES
017370  104050                          HLT     !DS!WC          ;SC AND TRE SHOULD SET IN RSCS1
                                                                ;BECAUSE OF ERROR IN RSER
017372  104414                  MROPIA: CLRDK                   ;CLEAR ALL ERRORS
017374  005777  161514                  TST     @RSER           ;DID OPI CLEAR IN RSER ?
017400  001437                          BEQ     1$              ;YES
017402  104402  017406                  TYPE    .+2             ;.ASCIZ <15><12>"OPI IN RSER DID NOT CLEAR BY SETTING CL
017476  104040                          HLT     !DS             ;RSER SHOULD=0
017500  022777  010600  161404  1$:     CMP     #10600,@RSDS    ;DID ERROR BITS CLEAR IN RSDS
                                                                ;BY SETTING CLR BIT IN RSCS2
017506  001401                          BEQ     .+4             ;YES
017510  104040                          HLT     !DS             ;RSDS SHOULD=10600
```

```
3879                                    ;*************************************************************************
3880                                    ;TEST 64                    PARITY ERROR TEST
3881                                    ;*************************************************************************
3882  017512  104400                    TST64:  SCOPE
3883
3884                                    ;MODULES TESTED: M7754, M7770
3885                                    ;SET "PAT" BIT IN RSCS2.  WRITE A DRIVE REGISTER.  "PAR" SHOULD SET IN
3886                                    ;THE DRIVE ERROR REGISTER (RSER) WHICH SHOULD CAUSE "ATA" TO SET IN RSAS
3887                                    ;AND 'SC' TO SET IN RSCS1.
3888
3889  017514  104414                    MRPAR:  CLRDK                   ;CLEAR ALL REGISTERS
3890  017516  042737  000040  001170            BIC     #BIT5,ONCEE     ;CLEAR CLK CNT FLAG
3891  017524  104416                            MRDMD                   ;PUT DRIVE IN MAINT MODE
3892  017526  104420                            MRCK                    ;CHECK MAINT TO
3893  017530  022701                            22701                   ;EQUAL 22701
3894  017532  104424                            MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
3895  017534  052777  000020  161340            BIS     #BIT4,@RSCS2    ;SET THE "PAT" BIT.
3896  017542  012777  000077  161340            MOV     #77,@RSDA       ;BY WRITING INTO THIS REGISTER,
3897                                                                    ;PAR SHOULD SET IN RSER
3898  017550  022777  000010  161336            CMP     #10,@RSER       ;DID PAR SET?
3899  017556  001401                            BEQ     .+4             ;YES
3900  017560  104040                            HLT     !DS             ;"PAR" IN RSER SHOULD BE SET FOR
3901                                                                    ;THE "PAT" BIT WAS SET IN RSCS2
3902                                                                    ;WHEN PROGRAM TRIED TO WRITE INTO RSDA
3903  017562  022777  104200  161310            CMP     #104200,@RSCS1  ;DID PAR CAUSE SC TO SET?
3904  017570  001401                            BEQ     .+4             ;YES
3905  017572  104044                            HLT     !DS!DA          ;SC SHOULD BE SET IN RSCS1 FOR
3906                                                                    ;PAR SHOULD BE SET IN RSER
3907  017574  022777  000077  161306            CMP     #77,@RSDA       ;DID RSDA GET LOADED?
3908  017602  001401                            BEQ     .+4             ;YES
3909  017604  104004                            HLT     !DA             ;RSDA SHOULD=77 FOR PAT
3910                                                                    ;BIT WAS SET WHEN PROGRAM
3911                                                                    ;TRIED TO WRITE INTO RSDA
3912  017606  104414                    CLRDK                           ;CLEAR ALL ERRORS
3913  017610  022777  004200  161262            CMP     #4200,@RSCS1    ;DID ERRORS CLEAR?
3914  017616  001401                            BEQ     .+4             ;YES
3915  017620  104044                            HLT     !DS!DA          ;SC DID NOT CLEAR BY USING
3916                                                                    ;THE "CLR" BIT IN RSCS2
3917  017622  005777  161266                    TST     @RSER           ;DID PAR CLEAR?
3918  017626  001401                            BEQ     .+4             ;YES
3919  017630  104044                            HLT     !DS!DA          ;PAR DID NOT CLEAR BY USING
3920                                                                    ;THE CLR BIT IN RSCS2
```

```
                                ;*****************************************************************************
                                ;TEST 65                    MAINTENANCE MODE INTERRUPT TEST
                                ;*****************************************************************************
017632  104400                  TST65:  SCOPE

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

017634  012737  001602  001144          MOV     #1602,FLAG2
017642  104414          MREX:   CLRDK                   ;CLEAR DRIVE REGISTERS
017644  012737  000200  177776          MOV     #200,@#PS       ;SETUP FOR INTERRUPT
017652  012706  000500                  MOV     #500,SP
017656  012737  000040  001170          MOV     #40,ONCEE       ;SET TYPE CLOCK CNT WITH ERROR MESSAGE FLAG
017664  104430                  MRIND                   ;SEND INDEX PULSE TO MR REG
017666  104420                  MRCK                    ;CHECK MR REG
017670  022701                  22701                   ;TO EQUAL 22701
017672  104424                  MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                        ;BY SENDING 2 CLOCK PULSES

                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (2 SECTORS)
                                ;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)

017674  012702  030114                  MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
017700  005022                  CLR     (R2)+           ;CLEAR 1ST LOCATION
017702  012722  177777          MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
017706  005003                  CLR     R3              ;CLEAR WORK LOC TO GENERATE
017710  000261                  SEC                     ;A PATTERN OF FLOATING ONES
017712  006103          1$:     ROL     R3              ;GET PATTERN
017714  103402                  BCS     2$              ;DONE GET OUT
017716  010322                  MOV     R3,(R2)+        ;FILL BUFFER
017720  000774                  BR      1$              ;CONT
017722  012703  000156  2$:     MOV     #110.,R3        ;FILL REMAINING PORTION OF
017726  012704  146314          MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
017732  010422          3$:     MOV     R4,(R2)+        ;LOAD BUFFER
017734  005303                  DEC     R3              ;DONE YET?
017736  001375                  BNE     3$              ;NO

                                ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (2 SECTORS)
017740  012777  020560  161162          MOV     #INTMR,@RSVEC   ;SETUP INTERRUPT VECTOR
017746  012777  000340  161156          MOV     #340,@RSVCPS
017754  012777  030114  161124          MOV     #INBUF,@RSBA    ;LOAD BUS ADDR REG
017762  012777  177600  161114          MOV     #177600,@RSWC   ;LOAD WORD COUNT REG
017770  012777  000161  161102          MOV     #161,@RSCS1     ;LOAD WRITE COMMAND I/E
017776  104454                  GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                        ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                        ;THAT OTHERWISE COULD NOT BE CLEARED.
020000  104220                  HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
020002  104456                  SPASS                   ;CLOCK MR REG SP = 1
```

```
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
020004  104430                                MRIND
020006  104420                                MRCK                    ;CHECK MR REG TO EQUAL
020010  020501                                20501                   ;20501
020012  104000                                HLT

                                        ;STEP THRU RESYNC PERIOD

020014  012737  001000  001204                MOV     #512.,REPT
020022  052737  000040  001170                BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
020030  104446                        MREX1:  MCLK1                   ;CLOCK MR REG
020032  104420                                MRCK                    ;CHECK FOR
020034  030511                                30511                   ;CORRECT DATA
020036  104000                                HLT                     ;MR = BAD GOOD = CORRECT DATA
020040  104450                                MCLK0                   ;CLOCK MR REG
020042  104420                                MRCK                    ;CHECK FOR
020044  020501                                20501                   ;CORRECT DATA
020046  104000                                HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
020050  005337  001204                        DEC     REPT            ;FINISH LOOPING
020054  001365                                BNE     MREX1           ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
020056  104446                                MCLK1                   ;CLOCK MR REG
020060  104420                                MRCK                    ;MR SHOULD
020062  030111                                30111                   ;EQUAL 30111
020064  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
020066  104450                                MCLK0                   ;CLOCK MR REG
020070  104420                                MRCK                    ;CHECK MR
020072  020101                                20101                   ;TO EQUAL 20101
020074  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;PERFORM 63 MAINT CLOCK OPERATIONS--WRITING PREAMBLE

020076  012737  000077  001204                MOV     #63.,REPT
020104  104446                        MREX2:  MCLK1                   ;CLOCK MR REG
020106  104420                                MRCK                    ;CHECK MR REG
020110  031511                                31511                   ;TO EQUAL 31511
020112  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
020114  104450                                MCLK0                   ;CLOCK MR REG
020116  104420                                MRCK                    ;CHECK MR REG
020120  021501                                21501                   ;TO EQUAL 21501
020122  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
020124  005337  001204                        DEC     REPT            ;DONE YET
020130  001365                                BNE     MREX2           ;NO LOOP
```

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

020132  104446                        MCLK1                   ;CLOCK MR REG
020134  104420                        MRCK                    ;CHECK MR REG
020136  131511                        131511                  ;TO EQUAL 131511
020140  104000                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
020142  104450                        MCLK0                   ;CLOCK MR REG
020144  104420                        MRCK                    ;MR REG SHOULD
020146  025501                        25501                   ;EQUAL 25501
020150  104000                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
020152  104446                        MCLK1
020154  104420                        MRCK
020156  135511                        135511
020160  104000                        HLT
                                ;PERFORM NEXT STEP 8 TIMES TO FINISH WRITING PREAMBLE
020162  012737  000010  001204        MOV     #10,REPT
020170  104450                MREX3:  MCLK0                   ;CLOCK MR REG
020172  104420                        MRCK                    ;CHECK MR REG
020174  025501                        25501                   ;TO EQUAL 25501
020176  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS
020200  104446                        MCLK1                   ;CLOCK MR REG
020202  104420                        MRCK                    ;CHECK MR REG
020204  135511                        135511                  ;TO EQUAL 135511
020206  104000                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
020210  005337  001204                DEC     REPT            ;DONE YES?
020214  001365                        BNE     MREX3           ;NO LOOP BACK

                                ;MOVE DATA WORD INTO RS03 SHIFT REGISTER

020216  104450                        MCLK0                   ;CLOCK MR REG
020220  104420                        MRCK                    ;CHECK MR REG
020222  021501                        21501                   ;TO EQUAL 21501
020224  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS
020226  104446                        MCLK1                   ;CLOCK MR REG
020230  104420                        MRCK                    ;MR REG SHOULD
020232  123511                        123511                  ;EQUAL 123511
020234  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS

                                ;ENCODE SYNC 1

020236  104450                        MCLK0                   ;CLOCK MR REG
020240  104420                        MRCK                    ;MR REG SHOULD NOW
020242  033501                        33501                   ;EQUAL 33501
020244  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS
020246  012705  030114                MOV     #INBUF,R5       ;GET STARTING ADDR FOR DATA BUFFER
020252  011504                        MOV     (R5),R4         ;GET DATA
```

```
4066  020254  012737  002156  001216          MOV     #1134.,WORK     ;DOING A 1 SECTOR TRANSFER 127 WORDS
4067                                                                    ;18 BITS PER WORD-CLOCK LOOPS
4068                                                                    ;TAKE CARE OF 2 BITS AT A TIME
4069                                                                    ;64 TIMES 18 EQUALS 1134 LOOPS
4070                                                                    ;TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
4071  020262  052737  000100  001170          BIS     #BIT6,ONCEE     ;SET 1ST TRANSFER WORD FLAG
4072  020270  104432                  1$:     XBIT                    ;GET 1 BIT OF DATA
4073  020272  104434                          CLKD1                   ;SET MCLK
4074                                                                    ;AND CALCULATE MR REG
4075                                                                    ;FOR CORRECT DATA (MWDB)
4076  020274  104000                          HLT                     ;MR REG NOT CORRECT
4077  020276  104436                          CLKD0                   ;CLEAR MCLK TO
4078                                                                    ;COMPLETE TRANSFER OF 1 BIT
4079                                                                    ;CALCULATE CORRECT ANS FOR
4080                                                                    ;MR REG (MWDB)
4081  020300  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
4082  020302  032737  000200  001170          BIT     #BIT7,ONCEE     ;ON LAST WORD YET?
4083  020310  001015                          BNE     2$              ;YES
4084  020312  032737  000400  001170          BIT     #BIT8,ONCEE     ;ON CRC WORD YET?
4085  020320  001043                          BNE     3$              ;YES
4086  020322  005337  001216                  DEC     WORK            ;DONE WITH 63 WORDS?
4087  020326  001360                          BNE     1$              ;NO
4088
4089  020330  052737  000200  001170          BIS     #BIT7,ONCEE     ;SET LAST WORD FLAG
4090  020336  012737  000023  001216          MOV     #19.,WORK       ;SET UP TO TRANSFER LAST WORD
4091  020344  005337  001216          2$:     DEC     WORK            ;DONE YET
4092  020350  001347                          BNE     1$
4093
4094  020352  052737  000400  001170          BIS     #BIT8,ONCEE     ;SET TRANSFERRING CRC WORD
4095  020360  042737  000200  001170          BIC     #BIT7,ONCEE     ;CLEAR LAST WORD FLAG
4096
4097                                  ;GENERATE RMR ERROR BY ATTEMPTING TO WRITE RSER
4098                                  ;EXC SHOULD THEN BE ASSERTED
4099
4100  020366  012777  177777  160520          MOV     #-1,@RSER
4101  020374  004737  024430                  JSR     PC,GENCRC       ;GENERATE CRC WORD
4102                                                                    ;AND LEAVE IN "WORK"
4103  020400  012702  030114                  MOV     #INBUF,R2       ;GO TO END
4104  020404  062702  000200                  ADD     #200,R2         ;OF DATA BUFFER
4105  020410  013712  001216                  MOV     WORK,@R2        ;LOAD CRC WORD
4106  020414  010205                          MOV     R2,R5           ;RESET POINTER FOR
4107  020416  162705  000002                  SUB     #2,R5           ;R5 FOR CRC WD
4108  020422  012737  000023  001216          MOV     #19.,WORK       ;SETUP TO XFER CRC
4109  020430  005337  001216          3$:     DEC     WORK            ;DONE YET?
4110  020434  001315                          BNE     1$              ;NO
```

```
4111                                    ;EBL SHOULD NOW ASSERT AND CRC BE WRITTEN
4112
4113  020436  104446                        MCLK1                   ;CLOCK MR REG TO STEP THROUGH DEAD BAND AREA
4114  020440  104420                        MRCK                    ;CHECK MR REG
4115  020442  113511                        113511                  ;TO EQUAL 113511
4116  020444  104000                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
4117
4118                                    ;LOOP 17 TIMES
4119
4120  020446  012737  000017  001204        MOV     #17,REPT
4121  020454  104450                4$:     MCLK0                   ;CLOCK MR REG
4122  020456  104420                        MRCK                    ;CHECK MR REG
4123  020460  003501                        3501                    ;TO EQUAL 3501
4124  020462  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS
4125  020464  104446                        MCLK1                   ;CLOCK MR REG
4126  020466  104420                        MRCK                    ;CHECK MR REG
4127  020470  113511                        113511                  ;TO EQUAL 113511
4128  020472  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS
4129  020474  005337  001204                DEC     REPT            ;DONE LOOPING YET?
4130  020500  001365                        BNE     4$              ;NO
4131
4132                                    ;FINISH UP
4133
4134  020502  104450                        MCLK0                   ;CLOCK MR REG
4135  020504  104420                        MRCK                    ;CHECK MR REG
4136  020506  003501                        3501                    ;TO EQUAL 3501
4137  020510  104000                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
4138  020512  104446                        MCLK1                   ;CLOCK MR REG
4139  020514  104420                        MRCK                    ;CHECK MR REG
4140  020516  111511                        111511                  ;TO EQUAL 111511
4141  020520  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS
4142
4143                                    ;TRANSFER SHOULD NOW BE COMPLETE
4144
4145  020522  104450                        MCLK0                   ;CLOCK MR REG
4146  020524  104420                        MRCK                    ;CHECK MRE REG
4147  020526  001501                        1501                    ;TO EQUAL 1501
4148  020530  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS
4149  020532  104446                        MCLK1                   ;CLOCK MR REG
4150  020534  104420                        MRCK                    ;CHECK MR
4151  020536  012711                        12711                   ;TO EQUAL
4152  020540  104000                        HLT                     ;12711
4153  020542  104450                        MCLK0                   ;CLOCK MR
4154  020544  104420                        MRCK                    ;CHECK MR
4155  020546  002201                        2201                    ;TO EQUAL
4156  020550  104000                        HLT                     ;2201
4157  020552  000240                        NOP                     ;STALL FOR TIME
4158  020554  104050                        HLT     !WC!DS          ;SHOULD NEVER GET HERE
4159  020556  000424                        BR      INTMR1          ;BECAUSE DRIVE SHOULD HAVE INTERRUPTED,
4160                                                                ;CAUSING JUMP TO INTMR.
4161                                                                ;CHECK FOR ASSERTION OF FTS ATTN L.
```

```
                                        ;NOW TEST CONTROLLER
                
004162                                  
004163                                  
004164  020560  022777  144260  160312  INTMR:  CMP     #144260,@RSCS1  ;IS CS1 CORRECT?
004165  020566  001401                          BEQ     .+4             ;YES
004166  020570  104014                          HLT     !DA!WC          ;YES
004167  020572  022777  000001  160310  5$:     CMP     #1,@RSDA        ;IS RSDA CORRECT?
004168  020600  001401                          BEQ     .+4             ;YES
004169  020602  104004                          HLT     !DA             ;DA SHOULD = 1
004170  020604  022777  000004  160302          CMP     #4,@RSER        ;DID RMR SET IN RSER
004171  020612  001401                          BEQ     .+4             ;YES
004172  020614  104050                          HLT     !DS!WC          ;RSER SHOULD = 4
004173  020616  022777  000001  160264          CMP     #1,@RSDA        ;DOES RSDA=1
004174  020624  001401                          BEQ     .+4             ;YES
004175  020626  104004                          HLT     !DA             ;RSDA SHOULD=1
004176  020630  000240                  INTMR1: NOP                     ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 66                    DISK ADDRESS OVERFLOW TEST
                                ;*****************************************************************************
020632  104400                  TST66:  SCOPE

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

020634  104414                  MRAOE:  CLRDK                   ;CLEAR ALL REGISTERS
020636  012706  000500                  MOV     #500,SP         ;SETUP STACK POINTER
020642  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
020644  104420                          MRCK                    ;CHECK MAINT REG
020646  022701                          22701                   ;TO EQUAL 22701
020650  104424                          MRINT                   ;INITIALIZE MAINT REG BY SENDING
                                                                ;2 CLOCK PULSES (CLEAR MRSP)
020652  012777  007777  160230          MOV     #7777,@RSDA     ;SETUP DISK ADDRESS
020660  012777  177400  160216          MOV     #-400,@RSWC     ;SETUP FOR A 2 SECTOR TRANSFER
020666  012777  030714  160212          MOV     #OUTBUF,@RSBA   ;GET OUTPUT BUFFER

                                ;SETUP BUFFER WITH ALL ONES
020674  012705  030714                  MOV     #OUTBUF,R5      ;GET STARTING ADDRESS OF OUTBUF
020700  012737  000400  001204          MOV     #400,REPT       ;LOAD 2 SECTORS
020706  012725  177777          1$:     MOV     #-1,(R5)+       ;WITH WORDS
020712  005337  001204                  DEC     REPT            ;OF ALL ONES
020716  001373                          BNE     1$

020720  012777  000061  160152          MOV     #61,@RSCS1      ;LOAD WRITE COMMAND
020726  104430                          MRIND                   ;SET INDEX PULSE

                                ;SUPPLY CLOCKS TO STEP THROUGH A TRACK

020730  012737  000002  001204          MOV     #2,REPT
020736  012704  124000          5$:     MOV     #43008.,R4      ;SETUP FOR FAST CLOCK PULSES 172032 CLOCKS
020742  012702  000011                  MOV     #11,R2
020746  012703  000001                  MOV     #1,R3
020752  010277  160146          2$:     MOV     R2,@RSMR
020756  010377  160142                  MOV     R3,@RSMR
020762  005304                          DEC     R4
020764  001372                          BNE     2$
020766  005337  001204                  DEC     REPT
020772  001361                          BNE     5$

020774  104422                          MRCLK                   ;CLOCK A 11 AND A 1 INTO RSMR
020776  104426                          DSCK                    ;CHECK MR
021000  012400                          12400                   ;TO EQUAL 12400
021002  104000                          HLT                     ;LBT SHOULD BE SET IN RSDS
```

```
                                        ;ASSERT MAINTENANCE INDEX PULSE TO RESET DRIVE
                                        ;FOR THE SECOND REVOLUTION
021004  104430                                MRIND                   ;ASSERT MAINT INDEX PULSE
021006  005037  001176                        CLR     MCCNT           ;CLEAR THE CLOCK COUNTER
021012  104420                                MRCK                    ;CHECK MR REG
021014  002501                                2501                    ;TO EQUAL 2501. SHOULD STILL BE WRITING
021016  104000                                HLT

                                        ;SUPPLY ENOUGH CLOCKS TO STEP THROUGH THE RS03 RESYNC PERIOD
021020  012737  001000  001204                MOV     #512.,REPT      ;CLOCK COUNT TO STEP THRU RESYNC
021026  104446                        4$:     MCLK1                   ;2ND REVOLUTION
021030  104420                                MRCK                    ;CHECK MR
021032  012511                                12511                   ;TO EQUAL 12511
021034  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
021036  104450                                MCLKO                   ;CLOCK MR REG
021040  104420                                MRCK                    ;CHECK MR
021042  002501                                2501                    ;REG TO
021044  104000                                HLT                     ;EQUAL 2501
021046  005337  001204                        DEC     REPT
021052  001365                                BNE     4$              ;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
021054  104422                        AOECK:  MRCLK
021056  104422                                MRCLK                   ;CAUSE SECTOR PULSE AND AOE ERROR
021060  104420                                MRCK                    ;CHECK FOR SECTOR PULSE
021062  022701                                22701                   ;IN RSMR
021064  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
021066  022777  001000  160020                CMP     #1000,@RSER     ;DID AOE SET IN RSER?
021074  001401                                BEQ     1$              ;AOE SHOULD BE SET IN RSER
021076  104040                                HLT     !DS             ;RSER SHOULD EQUAL 1000
021100  022777  152600  160004        1$:     CMP     #152600,@RSDS   ;IS RSDS CORRECT
021106  001401                                BEQ     2$              ;YES
021110  104040                                HLT     !DS             ;ERR & ATA SHOULD BE SET IN RSDS
                                                                        ;BECAUSE OF AOE ERROR IN RSER
021112  104414                        2$:     CLRDK                   ;CLEAR ERROR
021114  005777  157774                        TST     @RSER           ;DID ERROR CLEAR?
021120  001401                                BEQ     3$              ;YES
021122  104040                                HLT     !DS             ;AOE DID NOT CLEAR BY SETTING CLR IN RSCS2
021124  022777  010600  157760        3$:     CMP     #10600,@RSDS    ;DID ERRORS CLEAR
021132  001401                                BEQ     .+4             ;YES
021134  104040                                HLT     !DS             ;ERR AND ATA & LBT SHOULD ALL BE CLEARED
                                                                        ;FOR CLR WAS SET IN RSCS2
```

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

                                ;*************************************************************************
                                ;TEST 67                  MAINTENANCE MODE VERIFY TEST
                                ;*************************************************************************
021136  104400                  TST67:  SCOPE

                                ;MODULE TESTED G182

021140  032777  004000  157660  MRVR:   BIT     #BIT11,@SWR     ;DO THIS TEST?
021146  001002                  BNE     3$              ;YES
021150  000137  021660          JMP     @#INFTST        ;NO
021154  012737  001600  001144  3$:     MOV     #1600,FLAG2     ;SET VERIFY TEST FLAG
021162  104414                  CLRDK                   ;CLEAR ALL DRIVES
021164  012737  017777  001232          MOV     #17777,WORK5    ;STALL TO RESYNC
021172  005337  001232          4$:     DEC     WORK5           ;DRIVE
021176  001375                  BNE     4$              ;TIMING LOGIC

                                ;STEP THRU RESYNC PERIOD

021200  012777  170000  157676          MOV     #-10000,@RSWC   ;WRITE ONE TRACK - 4K WORDS
021206  012737  177777  030114          MOV     #177777,INBUF   ;WRITE A PATTERN 12525
021214  052777  000010  157660          BIS     #BIT3,@RSCS2    ;SET BAI BIT
021222  012777  030114  157656          MOV     #INBUF,@RSBA    ;SET DATA WD
021230  012737  007777  001204          MOV     #7777,REPT      ;SETUP WAIT LOOP
021236  012777  000061  157634          MOV     #61,@RSCS1      ;GO WRITE
021244  105777  157630          1$:     TSTB    @RSCS1          ;DONE YET?
021250  100404                  BMI     2$              ;YES
021252  005337  001204          DEC     REPT            ;DECREMENT COUNTER WAITING
021256  001372                  BNE     1$              ;FOR READY
021260  104000                  HLT                     ;READY NEVER CAME UP
021262  005777  157612          2$:     TST     @RSCS1          ;ANY ERRORS?
021266  100002                  BPL     MRVR1           ;NO
021270  104050                  HLT     !DS!WC          ;STOP HERE TILL THIS PROBLEM IS FIXED TRY DZRSB DIAG
021272  000433                  BR      TBDIA           ;TYPE MESSAGE
```

```
4315  021274  104414                    MRVR1:  CLRDK                   ;CLEAR ALL REGISTERS
4316  021276  012777  170000  157600            MOV     #-10000,@RSWC   ;SETUP WC
4317  021304  052777  000010  157570            BIS     #BIT3,@RSCS2    ;SET BAI
4318  021312  012777  030114  157566            MOV     #INBUF,@RSBA    ;SETUP RSBA
4319  021320  012737  007777  001204            MOV     #7777,REPT      ;SETUP WAIT LOOP
4320  021326  012777  000051  157544            MOV     #51,@RSCS1      ;DO A WRITE CHECK TO VERIFY DISK
4321  021334  105777  157540              1$:   TSTB    @RSCS1          ;TEST
4322  021340  100404                            BMI     2$              ;FOR READY TO COME BACK
4323  021342  005337  001204                    DEC     REPT            ;WAIT
4324  021346  001372                            BNE     1$              ;
4325  021350  104000                            HLT                     ;READY NEVER CAME BACK
4326  021352  005777  157522              2$:   TST     @RSCS1          ;ANY ERRORS?
4327  021356  100032                            BPL     MRVRR           ;NO
4328  021360  104050                            HLT     !DS!WC          ;STOP HERE WC FAILED
4329                                                                    ;GO TO DZRSB DIAG
4330                                                                    ;BEFORE TRYING TO DEBUG
4331                                                                    ;THIS TEST
4332  021362                          TBDIA:
4333  021362  104402  021366                    TYPE    .+2             ;.ASCIZ <15><12>"FAILED VERIFY TEST --- RUN DZRSB DIAGNO
4334  021444  000137  012320          MRVRR:  JMP     @#1RWRT         ;GO WRITE IN MAINTENANCE MODE
4335                                  ;NOW CHECK TO SEE IF DRIVE WAS WRITTEN ON IN MAINTENANCE MODE
4336
4337  021450  104414                    MRVR2:  CLRDK                   ;CLEAR ALL REGISTERS
4338  021452  012737  007777  001216            MOV     #7777,WORK      ;STALL
4339  021460  005337  001216              3$:   DEC     WORK            ;WAITING FOR
4340  021464  001375                            BNE     3$              ;DRIVE TO GET IN SYNC WITH INDEX PULSE
4341  021466  012777  170000  157410            MOV     #-10000,@RSWC   ;SETUP WC FOR 1 TRACK
4342  021474  052777  000010  157400            BIS     #BAI,@RSCS2     ;SET BAI
4343  021502  012777  030114  157376            MOV     #INBUF,@RSBA    ;SETUP RSBA
4344  021510  012737  177777  030114            MOV     #177777,INBUF   ;SETUP FOR COMPARE
4345  021516  012777  000051  157354            MOV     #51,@RSCS1      ;DO A WRITE CHECK
4346  021524  105777  157350              1$:   TSTB    @RSCS1          ;TEST FOR
4347  021530  100375                            BPL     1$              ;READY TO COME BACK
4348  021532  032777  040000  157342            BIT     #WCE,@RSCS2     ;DID WCE SET?
4349  021540  001442                            BEQ     2$              ;NO
4350  021542  104402  021546                    TYPE    .+2             ;.ASCIZ <15><12> "WRITE AMPLIFIER DID NOT GET DISABLED B
4351  021642  104040                            HLT     !DS
4352  021644  000404                            BR      4$              ;GET OUT
4353  021646  005777  157226              2$:   TST     @RSCS1          ;ANY ERRORS?
4354  021652  100001                            BPL     .+4             ;NO
4355  021654  104040                            HLT     !DS             ;SHOULD NOT HAVE ANY ERRORS HERE
4356  021656  000240                      4$:   NOP
4357                                                                    ;TRY DZRSB DIAGNOSTIC
4358
4359  021660  052737  000001  001170  INFTST: BIS     #BIT0,ONCEE     ;SET FOUND DRIVE FLAG
4360  021666  000137  002352                    JMP     @#TRYNX         ;GET NEXT DRIVE
```

```
4361                                    .SBTTL          $DONE - BELL AND SCOPE ROUTINE
4362
4363  021672  104400                    DONE:   SCOPE                   ;TERMINATIONG SCOPE FOR LOOPING
4364  021674  062737  000001  001006            ADD     #1,PCNT+2       ;ADD 1 TO THE PASS COUNT
4365  021702  005537  001004                    ADC     PCNT            ;MAKE IT DOUBLE PREC.
4366  021706  032777  002000  157112            BIT     #SW10,@SWR      ;RING THE BELL?
4367  021714  001004                            BNE     4$              ;NO!
4368  021716  104402  021722                    TYPE    .+2             ;.ASCIZ <BELL><177>
4369  021726  013700  000042            4$:     MOV     @#42,R0         ;GET MONITOR ADDRESS
4370  021732  001405                            BEQ     $END1           ;IF NONE
4371  021734  000005                            RESET
4372  021736  004710                    $ENDAD: JSR     7,(0)           ;GO TO MONITOR
4373  021740  000240  000240  000240            240,240,240             ;SAVE ROOM FOR ACT11
4374  021746  000137  021754            $END1:  JMP     MULSYS          ;RETURN
4375
4376  021752  000000                    .TBIT:  0                       ;T BIT FLAG
4377
4378                                    ;MULTI DRIVE SYSTEM?
4379
4380  021754                            MULSYS:
4381  021754  104402  021760                    TYPE    .+2             ;.ASCIZ <15><12>"END OF PASS"
4382  021776  005037  001010                    CLR     LAD
4383  022002  005037  001000                    CLR     ICNT
4384  022006  032737  000020  001146            BIT     #BIT4,FLAG3     ;MULTI DRVIE?
4385  022014  001002                            BNE     1$              ;NO
4386  022016  000137  002006                    JMP     @#MULTII        ;YES
4387  022022  000137  002564            1$:     JMP     @#NOWGO         ;TEST ONLY ONE DRIVE
4388
4389                                    ;ERROR TYPEOUT ROUTINE FOR NO-OP TEST
4390
4391  022026  032737  000004  001170    NOPERR: BIT     #BIT2,ONCEE     ;WERE WE HERE BEFORE?
4392  022034  001031                            BNE     1$              ;YES
4393  022036  052737  000004  001170            BIS     #BIT2,ONCEE     ;SET BEEN HERE BEFORE FLAG
4394  022044  104402  022050                    TYPE    .+2             ;.ASCIZ <15><12>"ERROR CAUSED BY NO-OP FUNCTION "
4395  022112  013746  001216                    MOV     WORK,-(6)       ;PUT WORK ON STACK
4396  022116  104406                            TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
4397  022120  000207                    1$:     RTS     PC
```

```
4398  022122  104402  022136          CHG:    TYPE    ,REGCHG          ;TYPE MESSAGE
4399  022126  013746  001216                  MOV     WORK,-(6)        ;PUT WORK ON STACK
4400  022132  104406                          TYPES                    ;TYPE STACK IN OCTAL - SUPRESS
4401  022134  000207                          RTS     PC
4402
4403  022136  044103  047101  042507  REGCHG: .ASCIZ  "CHANGED WITH NO-OP FUNCTION "
4404  022144  020104  044527  044124
4405  022152  047040  026517  050117
4406  022160  043040  047125  052103
4407  022166  047511  020116     000
4408
4409  022173     015  051012  051115  TRMR:   .ASCIZ  <15><12>"RMR  DID NOT SET BY WRITING INTO "
4410  022200  020040  044504  020104
4411  022206  047516  020124  042523
4412  022214  020124  054502  053440
4413  022222  044522  044524  043516
4414  022230  044440  052116  020117
4415  022236     000
4416          022240                  .EVEN
4417
4418  022240  104422                  .MRINT: MRCLK                    ;CLOCK THE MAINT REG WITH A 11 AND A 1
4419  022242  104422                          MRCLK                    ;SAME
4420  022244  000002                          RTI                      ;RETURN
4421
4422  022246  012777  000011  156650  .MRCLK: MOV     #11,@RSMR        ;CLOCK THE
4423  022254  012777  000001  156642          MOV     #1,@RSMR         ;MAINT REG
4424  022262  062737  000001  001200          ADD     #1,MCCNT+2       ;ADD 1 TO CLOCK COUNT
4425  022270  005537  001176                  ADC     MCCNT            ;MAKE DOUBLE PRECISION
4426  022274  000002                          RTI
4427
4428  022276  017700  156622          .MRCK:  MOV     @RSMR,BAD        ;GET THE CONTENTS OF RSMR
4429  022302  017601  000000                  MOV     @(SP),GOOD       ;GET THE CORRECT ANSWER
4430  022306  062716  000002                  ADD     #2,(SP)          ;UPDATE THE RETURN ADDRESS FOR AN ERROR
4431  022312  020100                          CMP     GOOD,BAD         ;IS THE MR REG CORRECT?
4432  022314  001002                          BNE     1$               ;NO EXIT
4433  022316  062716  000002                  ADD     #2,(SP)          ;UPDATE RETURN ADDRESS TO SKIP THE HLT FOR CORRECT ANS
4434  022322  000002                  1$:     RTI                      ;RETURN
4435
4436                                  ;SEND INDEX PULSE TO THE MAINTENANCE REGISTER
4437  022324  012777  000021  156572  .MRIND: MOV     #21,@RSMR        ;SEND INDEX
4438  022332  012777  000001  156564          MOV     #1,@RSMR         ;PULSE TO MR REG
4439  022340  000002                          RTI
4440  022342  017700  156544          .DSCK:  MOV     @RSDS,BAD        ;GET THE CONTENTS OF RSDS
4441  022346  017601  000000                  MOV     @(SP),GOOD       ;GET THE CORRECT ANS
4442  022352  062716  000002                  ADD     #2,(SP)          ;UPDATE THE RETURN ADDR FOR AN ERROR
4443  022356  020100                          CMP     GOOD,BAD         ;IS RSDS CORRECT
4444  022360  001002                          BNE     1$               ;NO EXIT
4445  022362  062716  000002                  ADD     #2,(SP)          ;UPDATE RETURN ADDRTO SKIP THE HLT  FOR CORRRECT ANS
4446  022366  000002                  1$:     RTI
```

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

022370  032737  000200  001144  .XBIT:  BIT     #BIT7,FLAG2     ;1ST 1 BIT OF 1ST WD?
022376  001446                          BEQ     2$              ;NO
022400  012737  000001  001152          MOV     #1,LSTOD        ;YES SETUP SYNC 1 BIT FOR END OF PREAMBLE;
                                                                ;TO CALCULATE BOTTOM BIT
022406  032737  000100  001170          BIT     #BIT6,ONCEE     ;1ST TIME THROUGH?
022414  001006                          BNE     5$              ;YES
022416  042737  000200  001144          BIC     #BIT7,FLAG2
022424  012737  000000  001152          MOV     #0,LSTOD
022432  042737  000100  001170  5$:     BIC     #BIT6,ONCEE     ;CLEAR 1ST TIME THROUGH FLAG
022440  005037  001210          4$:     CLR     CLKCNT          ;CLEAR CLOCK COUNTER AT START OF EACH WD
022444  032737  000400  001170          BIT     #BIT8,ONCEE     ;ON CRC WD?
022452  001062                          BNE     1$              ;YES
022454  005037  001156                  CLR     NOWOD           ;NO BITS 16 & 17 ARE 0

022460  032737  000400  001144          BIT     #BIT8,FLAG2     ;XFERING BIT 17?
022466  001003                          BNE     7$              ;YES
022470  042737  001000  001144          BIC     #BIT9,FLAG2     ;CLEAR FLAG FOR BIT 16
022476  042737  000400  001144  7$:     BIC     #BIT8,FLAG2     ;CLEAR FLAG FOR BIT 17
022504  012737  000020  001226  6$:     MOV     #16.,WORK3      ;LOOP 16 TIMES 1 FOR EACH BIT
022512  000002                          RTI                     ;EXIT
022514  013737  001156  001152  2$:     MOV     NOWOD,LSTOD     ;SAVE LAST BIT XFERED
022522  032737  001000  001144          BIT     #BIT9,FLAG2
022530  001343                          BNE     4$
022532  005737  001226                  TST     WORK3           ;DONE WITH WD YET?
022536  001013                          BNE     3$              ;NO
022540  032737  002000  001144          BIT     #BIT10,FLAG2    ;ON BIT 16 OF CRC WD?
022546  001334                          BNE     4$              ;YES
022550  062705  000002                  ADD     #2,R5           ;UPDATE BUFFER WD
022554  011504                          MOV     (R5),R4         ;GET DATA WD
022556  052737  001400  001144          BIS     #1400,FLAG2     ;SET BITS 8 & 9 IN FLAG2
022564  000725                          BR      4$
022566  005037  001156          3$:     CLR     NOWOD           ;CLEAR PRESENT BIT
022572  032737  001000  001144          BIT     #BIT9,FLAG2     ;DID WE XFER BITS 16 & 17 YET?
022600  001317                          BNE     4$              ;NO
022602  000241                          CLC
022604  006104                          ROL     R4              ;GET NEXT DATA BIT
022606  006137  001156                  ROL     NOWOD           ;PUT IT INTO NOWOD
022612  005337  001226                  DEC     WORK3           ;KEEP COUNT OF BITS IN THE WORD
022616  000002                          RTI                     ;EXIT

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

022620  005037  001156          1$:     CLR     NOWOD           ;CLEAR PRESENT BIT
022624  006104                          ROL     R4              ;GET NEXT BIT
022626  006137  001156                  ROL     NOWOD           ;TO BE XFERED
022632  032737  002000  001144          BIT     #BIT10,FLAG2    ;DONE WITH BITS 16 & 17 YET?
022640  001321                          BNE     6$              ;YES
022642  052737  002000  001144          BIS     #BIT10,FLAG2    ;NO
022650  042737  001000  001144          BIC     #BIT9,FLAG2
022656  000002                          RTI                     ;EXIT
```

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

022660  104446                .CLKD1: MCLK1                   ;CLOCK MR REG WITH AN 11
022662  005003                        CLR     R3              ;CLEAR WORK LOCATION
022664  005737  001156                TST     NOWOD           ;TEST ODD BIT NOW BEING SENT FOR A 1 OR A 0
022670  001005                        BNE     TSTEVB          ;NOW TEST EVEN DATA BIT ON 1ST CLOCK
                                                              ;NOW BIT IS A 1 MWDB IS 0
022672  005737  001152        1$:     TST     LSTOD           ;TEST THE LAST ODD DATA BIT THAT WAS SENT
022676  001002                        BNE     TSTEVB          ;LAST ODD DATA BIT WAS A 1
                                                              ;MWDB IS A 0
022700  052703  010000        2$:     BIS     #BIT12,R3       ;SET MWDB FOR LATER COMPARE WITH MR REG

022704  012701  123511        TSTEVB: MOV     #123511,GOOD    ;GET CORRECT ANS
022710  050301                        BIS     R3,GOOD         ;FOR MR REG
022712  004737  024224                JSR     PC,MRCAL        ;DETERMINE STATE OF SB & LSR BITS
022716  017700  156202                MOV     @RSMR,BAD       ;GET CONTENTS OF MR REG
022722  020100                        CMP     GOOD,BAD        ;IS MR REG CORRECT?
022724  001002                        BNE     2$              ;NO TYPE OUT MR REG
022726  062716  000002                ADD     #2,(SP)         ;UPDATE RETURN ADDR FOR CORRECT ANS
022732  000002                2$:     RTI                     ;RETURN
```

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

022734  104450                  .CLKDO: MCLKD                           ;CLOCK MR REG
022736  005737  001156                  TST     NOWOD                   ;IS THE PRESENT DATA BIT A 1?
022742  001403                          BEQ     1$                      ;NO IT IS A 0
022744  052703  010000                  BIS     #BIT12,R3               ;SET MWDB FOR BIT BEING SENT IS A 1
022750  000402                          BR      4$
022752  042703  010000          1$:     BIC     #BIT12,R3               ;CLEAR MWDB FOR PRESENT BIT IS A 0
022756  012701  023501          4$:     MOV     #23501,GOOD             ;GET CORRECT ANS
022762  050301                          BIS     R3,GOOD                 ;FOR MR REG
022764  004737  024224                  JSR     PC,MRCAL                ;DETERMINE STATE OF SB & LSR BITS
022770  017700  156130                  MOV     @RSMR,BAD               ;GET CONTENTS OF MR REG
022774  020100                          CMP     GOOD,BAD                ;IS MR REG CORRECT?
022776  001002                          BNE     5$                      ;NO TYPEOUT ERROR
023000  062716  000002                  ADD     #2,(SP)                 ;UPDATE ADDR FOR CORRECT ANS
023004  000002                  5$:     RTI                             ;RETURN

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

023006  012737  023116  001216  CRCTYP: MOV     #CRCTAB,WORK            ;GET STARTING LOC OF IC TABLE
023014  012737  000001  001222          MOV     #1,WORK1                ;SETUP TO TEST FIRST CHIP
023022  033737  001222  001174  1$:     BIT     WORK1,SAVEE             ;WAS IT THIS BIT?
023030  001006                          BNE     2$                      ;YES TYPE IT
023032  062737  000006  001216          ADD     #6,WORK                 ;NO INDEX TABLE POINTER
023040  006137  001222                  ROL     WORK1                   ;SETUP TO TEST NEXT CHIP
023044  000766                          BR      1$                      ;NOW TES IT
023046  004777  156144          2$:     JSR     PC,@WORK                ;TYPE OUT CHIP
023052  104402  023056                  TYPE    .+2                     ;.ASCIZ " IN THE CRC REG SHOULD BE SET"
023114  000207                          RTS     PC
```

```
4557                                    ;TABLE FOR CRC TEST 2 TYPEOUT ROUTINE
4558
4559  023116  104402  023306    CRCTAB: TYPE    ,E302
4560  023122  000207                    RTS     PC
4561  023124  104402  023314            TYPE    ,E305
4562  023130  000207                    RTS     PC
4563  023132  104402  023322            TYPE    ,E307
4564  023136  000207                    RTS     PC
4565  023140  104402  023330            TYPE    ,E3010
4566  023144  000207                    RTS     PC
4567  023146  104402  023337            TYPE    ,E3012
4568  023152  000207                    RTS     PC
4569  023154  104402  023346            TYPE    ,E3015
4570  023160  000207                    RTS     PC
4571  023162  104402  023355            TYPE    ,E242
4572  023166  000207                    RTS     PC
4573  023170  104402  023363            TYPE    ,E245
4574  023174  000207                    RTS     PC
4575  023176  104402  023371            TYPE    ,E247
4576  023202  000207                    RTS     PC
4577  023204  104402  023377            TYPE    ,E2410
4578  023210  000207                    RTS     PC
4579  023212  104402  023406            TYPE    ,E2412
4580  023216  000207                    RTS     PC
4581  023220  104402  023415            TYPE    ,E2415
4582  023224  000207                    RTS     PC
4583  023226  104402  023424            TYPE    ,E192
4584  023232  000207                    RTS     PC
4585  023234  104402  023432            TYPE    ,E197
4586  023240  000207                    RTS     PC
4587  023242  104402  023440            TYPE    ,E1910
4588  023246  000207                    RTS     PC
4589  023250  104402  023447            TYPE    ,E1915
4590  023254  000207                    RTS     PC
4591
4592                                    ;CLOCK MR REG WITH A 0-1
4593
4594  023256  012777  000001  155640  .MCLKB: MOV     #1,@RSMR
4595  023264  012777  000011  155632          MOV     #11,@RSMR
4596  023272  062737  000001  001200          ADD     #1,MCCNT+2
4597  023300  005537  001176                  ADC     MCCNT
4598  023304  000002                          RTI
```

```
 4599  023306  031505  026460  000062  E302:    .ASCIZ  "E30-2"
 4600  023314  031505  026460  000065  E305:    .ASCIZ  "E30-5"
 4601  023322  031505  026460  000067  E307:    .ASCIZ  "E30-7"
 4602  023330  031505  026460  030061  E3010:   .ASCIZ  "E30-10"
 4603  023336      000
 4604  023337      105  030063  030455  E3012:   .ASCIZ  "E30-12"
 4605  023344  000062
 4606  023346  031505  026460  032461  E3015:   .ASCIZ  "E30-15"
 4607  023354      000
 4608  023355      105  032062  031055  E242:    .ASCIZ  "E24-2"
 4609  023362      000
 4610  023363      105  032062  032455  E245:    .ASCIZ  "E24-5"
 4611  023370      000
 4612  023371      105  032062  033455  E247:    .ASCIZ  "E24-7"
 4613  023376      000
 4614  023377      105  032062  030455  E2410:   .ASCIZ  "E24-10"
 4615  023404  000060
 4616  023406  031105  026464  031061  E2412:   .ASCIZ  "E24-12"
 4617  023414      000
 4618  023415      105  032062  030455  E2415:   .ASCIZ  "E24-15"
 4619  023422  000065
 4620  023424  030505  026471  000062  E192:    .ASCIZ  "E19-2"
 4621  023432  030505  026471  000067  E197:    .ASCIZ  "E19-7"
 4622  023440  030505  026471  030061  E1910:   .ASCIZ  "E19-10"
 4623  023446      000
 4624  023447      105  034461  030455  E1915:   .ASCIZ  "E19-15"
 4625  023454  000065
 4626
 4627                                  ;CLOCK MR REG WITH A 1
 4628
 4629  023456  012777  000011  155440  .MCLK1:  MOV     #11,@RSMR
 4630  023464  062737  000001  001200           ADD     #1,MCCNT+2
 4631  023472  005537  001176                   ADC     MCCNT
 4632  023476  000002                           RTI
 4633
 4634                                  ;CLOCK MR REG WITH A0
 4635
 4636  023500  012777  000001  155416  .MCLK0:  MOV     #1,@RSMR
 4637  023506  000002                           RTI
```

```
                                        ;GET ONE BIT OF DATA FROM INBUF
                                        ;FOR READING FROM DRIVE TO DETERMINE THE
                                        ;STATE OF MRDB IN THE MR REG.

023510  005737  001226        .RBIT:  TST     WORK3           ;STARTING NEW WD?
023514  001035                BNE     3$              ;NO
023516  062705  000002                ADD     #2,R5           ;UPDATE BUFFER WD
023522  011504                MOV     (R5),R4         ;GET DATA WD
023524  052737  004000  001144        BIS     #BIT11,FLAG2    ;SET TO INDICATE BIT 17
023532  005037  001210        5$:     CLR     CLKCNT          ;CLEAR CLOCK COUNTER AT START OF EACH WD
023536  032737  000400  001170        BIT     #BIT8,ONCEE     ;ON CRC WD?
023544  001041                BNE     1$              ;YES
023546  032737  000040  001144        BIT     #BIT5,FLAG2     ;IN CRC TEST ????
023554  001407                BEQ     7$              ;NO
023556  012737  000020  001226        MOV     #16.,WORK3
023564  042737  004000  001144        BIC     #BIT11,FLAG2    ;FOR CRC TEST
023572  000416                BR      4$
023574  005037  001156        7$:     CLR     NOWOD           ;BITS 16 + 17 OF DATA WORD ARE 0
023600  012737  000020  001226  6$:     MOV     #16.,WORK3      ;16 LOOPS FOR REMAINING 16 BITS OF WORD
023606  000002                RTI
023610  032737  004000  001144  3$:     BIT     #BIT11,FLAG2    ;IS THIS BIT 16?
023616  001404                BEQ     4$              ;NO
023620  042737  004000  001144        BIC     #BIT11,FLAG2    ;RTRANSFER BIT 16
023626  000741                BR      5$
023630  005037  001156        4$:     CLR     NOWOD           ;CLEAR PRESENT BIT
023634  006104                ROL     R4              ;GET NEXT DATA BIT
023636  006137  001156                ROL     NOWOD           ;SAVE IT IN ODD BIT
023642  005337  001226                DEC     WORK3           ;KEEP COUNT OF BITS IN THE WORD
023646  000002                RTI                     ;RETURN
                                        ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS, 0 & 1 ARE ALWAYS 0
023650  005037  001156        1$:     CLR     NOWOD           ;GET BITS 17
023654  006104                ROL     R4              ;AND 16
023656  006137  001156                ROL     NOWOD           ;FOR CRC WORD
023662  000746                BR      6$              ;CONTINUE
```

```
4672  023664  004737  024202            .CLKR1: JSR     PC,CALRTB       ;CALCULATE MRDB BIT FOR MR REG
4673  023670  012703  000011                    MOV     #11,R3          ;SETUP CLOCK BITS
4674  023674  062737  000001  001200            ADD     #1,MCCNT+2      ;INCREMENT
4675  023702  005537  001176                    ADC     MCCNT           ;CLOCK COUNT
4676  023706  053703  001216                    BIS     WORK,R3         ;SET BOTTOM BITS
4677  023712  010377  155206                    MOV     R3,@RSMR        ;SEND
4678  023716  012701  133611                    MOV     #133611,GOOD    ;CALCULATE CORRECT ANS FOR MR REG
4679  023722  032737  000004  001144            BIT     #BIT2,FLAG2     ;WRITE CK TEST?
4680  023730  001402                            BEQ     7$              ;NO
4681  023732  052701  000100                    BIS     #BIT6,GOOD      ;YES SET RD IN MR REG
4682  023736  050301                    7$:     BIS     R3,GOOD
4683  023740  032737  000400  001170            BIT     #BIT8,ONCEE     ;ON CRC WD?
4684  023746  001406                            BEQ     2$              ;NO
4685  023750  022737  000022  001204            CMP     #22,REPT        ;SHOULD CRCW BE SET?
4686  023756  001402                            BEQ     2$              ;YES
4687  023760  042701  020000                    BIC     #20000,GOOD     ;CLEAR CRCW
4688  023764  005337  001206            2$:     DEC     REPT1           ;SHOULD SB SET
4689  023770  001017                            BNE     6$              ;NO
4690  023772  012737  000044  001206            MOV     #44,REPT1       ;RESET SB COUNTER
4691  024000  052701  004000                    BIS     #BIT11,GOOD     ;SET SB
4692  024004  032737  000400  001170    3$:     BIT     #BIT8,ONCEE     ;ON CRC WD?
4693  024012  001406                            BEQ     6$              ;NO
4694  024014  022737  000043  001206            CMP     #43,REPT1       ;SHOULD SB AND CRCW BE SET ?
4695  024022  001002                            BNE     6$              ;NO
4696  024024  052701  020000                    BIS     #20000,GOOD     ;SET SB AND CRCW
4697  024030  017700  155070            6$:     MOV     @RSMR,BAD       ;GET MR REG
4698  024034  020100                            CMP     GOOD,BAD        ;IS RSMR CORRECT?
4699  024036  001002                            BNE     4$              ;NO
4700  024040  062716  000002                    ADD     #2,(SP)         ;YES
4701  024044  000002                    4$:     RTI                     ;RETURN
```

```
 4702  024046  053703  001216            .CLKRD: BIS     WORK,R3         ;SET BOTTOM BITS
 4703  024052  042703  000010                    BIC     #BIT3,R3
 4704  024056  010377  155042                    MOV     R3,@RSMR        ;SEND
 4705  024062  012701  023601                    MOV     #23601,GOOD     ;CALCULATE CORRECT ANS FOR MR REG
 4706  024066  032737  000004  001144            BIT     #BIT2,FLAG2     ;WRITE CK TEST?
 4707  024074  001402                            BEQ     7$              ;NO
 4708  024076  052701  000100                    BIS     #BIT6,GOOD      ;YES SET RD IN MR REG
 4709  024102  050301                    7$:     BIS     R3,GOOD
 4710  024104  032737  000400  001170            BIT     #BIT8,ONCEE     ;ON CRC WD?
 4711  024112  001402                            BEQ     2$              ;NO
 4712  024114  042701  020000                    BIC     #20000,GOOD     ;CLEAR CRCW
 4713  024120  005337  001206            2$:     DEC     REPT1           ;SHOULD SB SET?
 4714  024124  001017                            BNE     6$              ;NO
 4715  024126  012737  000022  001206            MOV     #18.,REPT1      ;RESET SB COUNTER
 4716  024134  052701  004000                    BIS     #BIT11,GOOD     ;SET SB
 4717  024140  032737  000400  001170    3$:     BIT     #BIT8,ONCEE     ;ON CRC WD?
 4718  024146  001406                            BEQ     6$              ;NO
 4719  024150  022737  000022  001206            CMP     #22,REPT1       ;SHOULD SB AND CRCW BE SET ?
 4720  024156  001002                            BNE     6$              ;NO
 4721  024160  052701  020000                    BIS     #20000,GOOD     ;SET SB AND CRCW
 4722  024164  017700  154734            6$:     MOV     @RSMR,BAD       ;GET MR REG
 4723  024170  020100                            CMP     GOOD,BAD        ;IS RSMR CORRECT?
 4724  024172  001002                            BNE     4$              ;NO
 4725  024174  062716  000002                    ADD     #2,(SP)         ;YES
 4726  024200  000002                    4$:     RTI                     ;RETURN
```

```
                                        ;CALCULATE THE STATE OF MRDB FROM CURRENT INPUT BIT
                                        ;LOCATION WORK CONTAINS CORRECT DATA FOR  MRDB
024202  005037  001216          CALRTB: CLR     WORK            ;CLEAR WORK LOCATION
024206  005737  001156                  TST     NOWOD           ;IS CURRENT BIT A 0?
024212  001403                          BEQ     2$              ;YES
024214  052737  000004  001216          BIS     #BIT2,WORK      ;NO SET MRDB
024222  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

024224  005237  001210          MRCAL:  INC     CLKCNT          ;ADD ONE TO CLOCK COUNT OF WORD
024230  032737  000200  001170          BIT     #BIT7,ONCEE     ;TRANSFERRING LAST WORD?
024236  001032                          BNE     LSTWD           ;YES
024240  032737  000400  001170          BIT     #BIT8,ONCEE     ;TRANSFERRING CRC WORD?
024246  001051                          BNE     CRCWD           ;YES
024250  022737  000016  001210          CMP     #16,CLKCNT      ;CLOCK COUNT 16 OR GREATER?
024256  101401                          BLOS    1$              ;YES
024260  000406                          BR      2$              ;GET OUT
024262  022737  000040  001210  1$:     CMP     #40,CLKCNT      ;CLOCK COUNT 40 OR GREATER?
024270  101402                          BLOS    2$              ;YES GET OUT
024272  052701  004000                  BIS     #BIT11,GOOD     ;SET SB BIT
024276  022737  000037  001210  2$:     CMP     #37,CLKCNT
024304  001404                          BEQ     3$
024306  022737  000040  001210          CMP     #40,CLKCNT
024314  001002                          BNE     4$
024316  042701  002000          3$:     BIC     #BIT10,GOOD     ;CLEAR LSR
024322  000207                  4$:     RTS     PC              ;RETURN

                                        ;CALCULATE MR FOR LAST DATA WORD
024324  022737  000036  001210  LSTWD:  CMP     #36,CLKCNT      ;IS THIS CLOCK 36 OR LESS?
024332  103016                          BHIS    2$              ;YES GETOUT
024334  022737  000037  001210          CMP     #37,CLKCNT      ;IS THIS CLOCK 15?
024342  001003                          BNE     3$              ;NO
024344  042701  002000          4$:     BIC     #BIT10,GOOD     ;YES CLEAR LSR
024350  000407                          BR      2$
024352  022737  000040  001210  3$:     CMP     #40,CLKCNT
024360  001001                          BNE     5$
024362  000770                          BR      4$
024364  042701  020000          5$:     BIC     #BIT13,GOOD     ;CLEAR CRCW BIT
024370  000207                  2$:     RTS     PC

                                        ;CALCULATE MR FOR CRC WORD

024372  042701  020000          CRCWD:  BIC     #BIT13,GOOD     ;CLEAR CRCW BIT
024376  022737  000037  001210          CMP     #37,CLKCNT      ;IS THIS CLOCK 17?
024404  001002                          BNE     2$              ;NO
024406  042701  002000                  BIC     #BIT10,GOOD     ;CLEAR LSR BIT
024412  022737  000040  001210  2$:     CMP     #40,CLKCNT
024420  001002                          BNE     1$
024422  042701  002000                  BIC     #BIT10,GOOD
024426  000207                  1$:     RTS     PC              ;RETURN
```

```
 4779
 4780                                   ;GENERATE A CRC WORD FROM THE DATA BUFFER
 4781                                   ;AND LEAVE THE CRC WORD IN "WORK" LOCATION
 4782                                   ;EXIT ROUTINE WITH RTS PC
 4783
 4784  024430  012737  000100  001204  GENCRC: MOV     #64.,REPT       ;64 WORDS PER SECTOR
 4785  024436  032737  000040  001144         BIT     #BIT5,FLAG2     ;IN CRC TEST?
 4786  024444  001403                         BEQ     13$             ;NO
 4787  024446  012737  000110  001204         MOV     #72.,REPT       ;YES
 4788  024454  012705  030114         13$:    MOV     #INBUF,R5       ;GET STARTING ADDR OF OUTPUT BUFFER
 4789  024460  011504                         MOV     (R5),R4         ;GET DATA WD
 4790  024462  005037  001220                 CLR     WORK0           ;CLEAR WORK LOCATION
 4791
 4792                                   ;INBIT CONTAINS PRESENT INPUT BIT
 4793                                   ;WK15 = BIT15 IF CRC AT TIME T
 4794                                   ;WORK0 = CRC AT TIME T + DURING FINAL MANIPULATION
 4795                                   ;WORK = BITS FROM SAVED CRC WORD (WCRC)
 4796
 4797  024466  012737  000022  001206  1$:     MOV     #18.,REPT1      ;GET 18 BITS PER WD
 4798  024474  032737  000040  001144         BIT     #BIT5,FLAG2     ;IN CRC TEST?
 4799  024502  001403                         BEQ     2$              ;NO
 4800  024504  012737  000020  001206         MOV     #16.,REPT1      ;YES
 4801  024512  013737  001220  001202  2$:     MOV     WORK0,WCRC      ;SAVE CURRENT CRC WD
 4802  024520  005037  001214                 CLR     WK15            ;CLEAR BIT 15 FROM CRC AT T 1
 4803  024524  000241                         CLC                     ;CLEAR CARY
 4804  024526  006137  001220                 ROL     WORK0           ;SHIFT CRC WD LEFT
 4805  024532  006137  001214                 ROL     WK15            ;CONTAINS BIT 15 OF CRC
 4806  024536  032737  000040  001144         BIT     #BIT5,FLAG2     ;IN CRC TEST?
 4807  024544  001004                         BNE     12$             ;YES
 4808  024546  022737  000021  001206         CMP     #17.,REPT1      ;DONE BITS 16 AND 17 YET?
 4809  024554  101406                         BLOS    3$              ;NO
 4810  024556  005037  001212         12$:    CLR     INBIT           ;CLEAR WORK LOC
 4811  024562  006104                         ROL     R4              ;PUT DATA BIT FROM BUFFER
 4812  024564  006137  001212                 ROL     INBIT           ;IN WORK1 LOC
 4813  024570  000402                         BR      4$
 4814  024572  005037  001212         3$:     CLR     INBIT           ;FOR BITS 16 AND 17
 4815  024576  013737  001214  001216  4$:     MOV     WK15,WORK       ;GET BIT 15 OF CRC
 4816  024604  004737  025030         5$:     JSR     PC,XXOR         ;XOR BIT15 WITH INPUT BIT
 4817  024610  042737  000001  001220         BIC     #BIT0,WORK0
 4818  024616  005737  001212                 TST     INBIT           ;TEST RESULT OF XOR
 4819  024622  001403                         BEQ     6$
 4820  024624  052737  000001  001220         BIS     #BIT0,WORK0
 4821  024632  013737  001212  001160  6$:     MOV     INBIT,RS0       ;SAVE XOR SESULT OF BIT 0 AND INPUT
```

```
                                        ;FROM B0 IN WORK0 AND B1 IN SAVED CRC (WCRC) CLACULATE
                                        ;NEW B2 FOR WORK0
                                
024640  005037  001216                  CLR     WORK
024644  032737  000002  001202          BIT     #BIT1,WCRC
024652  001403                          BEQ     7$
024654  052737  000001  001216          BIS     #BIT0,WORK
024662  013737  001160  001212  7$:     MOV     RS0,INBIT
024670  004737  025030                  JSR     PC,XXOR
024674  042737  000004  001220          BIC     #BIT2,WORK0
024702  005737  001212                  TST     INBIT           ;TEST RESULT OF XOR
024706  001403                          BEQ     8$
024710  052737  000004  001220          BIS     #BIT2,WORK0
                                
                                        ;FROM B0 IN WORK0 AND B14 IN WCRC CLACULATE BIT15 IN WORK0
                                
024716  005037  001216          8$:     CLR     WORK
024722  032737  040000  001202          BIT     #BIT14,WCRC
024730  001403                          BEQ     9$
024732  052737  000001  001216          BIS     #BIT0,WORK
024740  013737  001160  001212  9$:     MOV     RS0,INBIT
024746  004737  025030                  JSR     PC,XXOR
024752  042737  100000  001220          BIC     #BIT15,WORK0
024760  005737  001212                  TST     INBIT           ;TEST RESULT OF XOR
024764  001403                          BEQ     10$
024766  052737  100000  001220          BIS     #BIT15,WORK0
024774  005337  001206          10$:    DEC     REPT1           ;DONE WITH WD
025000  001244                          BNE     2$              ;NO
025002  005337  001204                  DEC     REPT            ;DONE WITH SECTOR?
025006  001404                          BEQ     11$             ;YES
025010  062705  000002                  ADD     #2,R5           ;GET NEXT WD
025014  011504                          MOV     (R5),R4         ;GET DATA WD
025016  000623                          BR      1$
025020  013737  001220  001216  11$:    MOV     WORK0,WORK      ;SAVE CRC WORD IN WORK
025026  000207                          RTS     PC              ;EXIT
                                
                                        ;XOR SUBROUTINE
                                
025030  013703  001216          XXOR:   MOV     WORK,R3
025034  043703  001212                  BIC     INBIT,R3
025040  043737  001216  001212          BIC     WORK,INBIT
025046  050337  001212                  BIS     R3,INBIT
025052  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.

025054  010446                  .TYPE:  MOV     R4,-(6)         ;SAVE R4
025056  010546                          MOV     R5,-(6)         ;SAVE R5
025060  017605  000004                  MOV     @4(6),R5        ;GET ADDRESS TO BE TYPED
025064  032705  177400                  BIT     #177400,R5      ;IS IT A TYPEM?
025070  001002                          BNE     1$              ;NO
025072  016605  000004                  MOV     4(6),R5         ;GET ADDRESS OF CHARACTER
025076  105715                  1$:     TSTB    (R5)            ;TERMINATOR?
025100  001423                          BEQ     2$              ;GET OUT IF SO
025102  122715  000012                  CMPB    #12,(R5)        ;IS THE CHAR A LINE FEED
025106  001012                          BNE     4$              ;NO - GET OUT
025110  113704  001015                  MOVB    FILCHR+1,R4     ;GET THE FILL COUNT
025114  113777  001014  153702  5$:     MOVB    FILCHR,@TPB     ;TYPE A FILLER
025122  105777  153670                  TSTB    @TPS            ;DONE YET?
025126  100375                          BPL     .-4             ;NO - WAIT
025130  005304                          DEC     R4              ;DEC COUNT
025132  001370                          BNE     5$              ;LOOP UNTIL 0
025134  112577  153664          4$:     MOVB    (R5)+,@TPB      ;LOAD AND TYPE THE CHARACTER
025140  105777  153652                  TSTB    @TPS            ;IS THE PRINTER READY
025144  100375                          BPL     .-4             ;WAIT UNTIL IT IS
025146  000753                          BR      1$              ;GET THE NEXT CHARACTER
025150  017646  000004          2$:     MOV     @4(6),-(6)      ;GET ADDRESS TO BE TYPED
025154  062766  000002  000006          ADD     #2,6(6)         ;ADD 2 TO THE ADDRESS
025162  022666  000004                  CMP     (6)+,4(6)       ;IS IT .+2?
025166  001006                          BNE     3$              ;NO
025170  062705  000002                  ADD     #2,R5           ;ADD 2 TO THE ADDRESS
025174  042705  000001                  BIC     #1,R5           ;BACK UP TO AN EVEN BYTE
025200  010566  000004                  MOV     R5,4(6)         ;RESTORE ADDRESS
025204  012605                  3$:     MOV     (6)+,R5         ;RESTORE R5
025206  012604                          MOV     (6)+,R4         ;RESTORE R4
025210  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:"

025212  104464                  .SCOPE: KBDIN                   ;GO CHECK FOR ↑G
025214  032777  000400  153604          BIT     #SW8,@SWR       ;LOOP ON SPEC. TEST?
025222  001404                          BEQ     1$              ;NO LOOP ON SPEC. TEST
025224  127737  153576  001000          CMPB    @SWR,ICNT       ;ON RIGHT TEST?   #SW7-0#
025232  001453                          BEQ     .OVER           ;NOT RIGHT TEST
025234  032777  040000  153564  1$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
025242  001045                          BNE     .KIT            ;LOOP ON TEST IS SET
025244  000416                          BR      3$              ;SKIP - NOP FOR XOR TESTER
025246  013746  000004                  MOV     @#4,-(6)        ;PUSH @#4 ON STACK
025252  012737  025272  000004          MOV     #4$,@#4         ;SET FOR TIMEOUT
025260  005737  177060                  TST     @#177060        ;ERROR ON XOR?
025264  012637  000004                  MOV     (6)+,@#4        ;POP STACK INTO @#4
025270  000422                          BR      .SVLAD          ;NO ERROR - GO TO NEXT TEST
025272  022626                  4$:     CMP     (6)+,(6)+       ;CLEAR STACK
025274  012637  000004                  MOV     (6)+,@#4        ;POP STACK INTO @#4
025300  000426                          BR      .KIT            ;ERROR - LOOP ON TEST
025302  032777  004000  153516  3$:     BIT     #SW11,@SWR      ;KILL ITERATIONS
025310  001012                          BNE     .SVLAD          ;YES - KILL ITERATIONS
025312  105737  001001                  TSTB    ICNT+1          ;FIRST ONE?
025316  001404                          BEQ     2$              ;BRANCH IF FIRST
025320  123737  025404  001001          CMPB    TIMES,ICNT+1    ;DONE?
025326  003013                          BGT     .KIT            ;BRANCH IF NOT
025330  112737  000001  001001  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
025336  105237  001000          .SVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
025342  011637  001010                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
025346  013777  001000  153454          MOV     ICNT,@DISPLAY   ;DISPLAY TEST NO. AND ITERATION COUNT
025354  000002                          RTI                     ;RETURN

025356  105237  001001          .KIT:   INCB    ICNT+1          ;INC THE ITERATION COUNT
025362  013777  001000  153440  .OVER:  MOV     ICNT,@DISPLAY   ;SET UP DISPLAY
025370  005737  001010                  TST     LAD             ;FIRST ONE?
025374  001760                          BEQ     .SVLAD          ;YES
025376  013716  001010                  MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
025402  000002                          RTI                     ;FIXES PS

025404  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.

025406  104464                  .HLT:   KBDIN                   ;GO CHECK FOR ^G
025410  032777  002000  153410          BIT     #SW10,@SWR      ;BELL ON ERROR?
025416  001402                  1$:     BEQ     1$              ;NO - SKIP
025420  104402  000007                  TYPE    ,BELL           ;RING BELL
025424  005237  001002                  INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
025430  032777  020000  153370          BIT     #SW13,@SWR      ;SKIP TYPEOUT IF SET
025436  001025                          BNE     2$              ;SKIP TYPEOUTS
025440  104402  025444                  TYPE    .+2             ;.ASCIZ <15><12>
025450  011637  001012                  MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
025454  162737  000002  001012          SUB     #2,HLTADR       ;FUDGE ADDRESS
025462  117737  153324  025544          MOVB    @HLTADR,.HLTCT  ;GET HLT ARGUEMENT
025470  013746  001012                  MOV     HLTADR,-(6)     ;PUT HLTADR ON STACK
025474  104404                          TYPEO                   ;TYPE STACK IN OCTAL
025476  104402  025502                  TYPE    .+2             ;.ASCIZ " "
025506  004737  027214                  JSR     PC,RSREG        ;GO TO USER ERROR ROUTINE
025512  005777  153310          2$:     TST     @SWR            ;HALT ON ERROR
025516  100001                          BPL     .+4             ;SKIP IF CONTINUE
025520  000000                          HALT                    ;HALT ON ERROR!
025522  032777  001000  153276          BIT     #SW9,@SWR       ;CHECK FOR INHIBIT LOOP ON ERROR
025530  001003                          BNE     3$              ;SKIP IF LOOP ON ERROR
025532  105037  001001                  CLRB    ICNT+1          ;CLEAR ITERATION COUNT
025536  000002                          RTI                     ;RETURN
025540  000137  025356          3$:     JMP     .KIT            ;LOOP ON TEST UNTIL NO ERRORS

025544  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.

025546  012737  170101  025734  .TYPEB: MOV     #170101,.PR     ;SET BIT FLAG AND 16. CHARACTER COUNT
025554  000411                  .TYPEO: BR      .PTIT           ;NOW TYPE IT IN BIT FORM
025556  112737  000001  025734  .TYPEO: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
025564  000402                          BR      .+6             ;SKIP
025566  005037  025734          .TYPES: CLR     .PR             ;SUPRESS LEADING ZERO'S
025572  112737  177772  025735          MOVB    #-6,.PR+1       ;SET COUNT
025600                          .PTIT:
025600  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
025602  010546                          MOV     R5,-(6)         ;PUSH R5 ON STACK
025604  016605  000010                  MOV     10(6),R5        ;GET THE DATA
025610  012704  025736                  MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
025614  105014                          CLRB    (4)             ;CLEAR FIRST BYTE
025616  000411                          BR      .PRF            ;ROTATE FIRST BIT
025620  105014                  .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
025622  032737  000100  025734          BIT     #100,.PR        ;BIT TYPING MODE?
025630  001004                          BNE     .PRF            ;YES - SKIP 2 ROTATES
025632  006105                          ROL     R5              ;ROTATE BIT INTO C
025634  106114                          ROLB    (4)             ;PACK IT
025636  006105                          ROL     R5              ;ROTATE BIT INTO C
025640  106114                          ROLB    (4)             ;PACK IT
025642  006105                  .PRF:   ROL     R5              ;ROTATE BIT INTO C
025644  106114                          ROLB    (4)             ;PACK IT
025646  105714                          TSTB    (4)             ;IS IT ZERO?
025650  001402                          BEQ     .+6             ;SKIP INC
025652  105237  025734                  INCB    .PR             ;SET FILL SWITCH
025656  105737  025734                  TSTB    .PR             ;CHECK FILL SWITCH
025662  001402                          BEQ     .+6             ;SKIP BITSET
025664  152724  000060                  BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
025670  105237  025735                  INCB    .PR+1           ;INC COUNT
025674  001351                          BNE     .PRL            ;REPEAT
025676  022704  025736                  CMP     #.PR+2,R4       ;EMPTY BUFFER?
025702  001002                          BNE     .+6             ;SKIP IF NOT
025704  112724  000060                  MOVB    #'0,(4)+        ;LOAD 1 ZERO
025710  105014                          CLRB    (4)             ;NULL TERMINATOR
025712  104402  025736                  TYPE    .PR+2           ;TYPE IT
025716  012605                          MOV     (6)+,R5         ;POP STACK INTO R5
025720  012604                          MOV     (6)+,R4         ;POP STACK INTO R4
025722  016666  000002  000004          MOV     2(6),4(6)       ;GET RID OF
025730  012616                          MOV     (6)+,(6)        ;DATA WORD
025732  000002                          RTI                     ;RETURN

025734  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'.

025760  012777  026106  000126  .POWER: MOV     #.ILLUP,@.PUVEC ;SET FOR FAST UP
025766  012777  000340  000122          MOV     #340,@.PUVEC+2  ;PRIO:7
025774  010046                        MOV     R0,-(6)         ;PUSH R0 ON STACK
025776  010146                        MOV     R1,-(6)         ;PUSH R1 ON STACK
026000  010246                        MOV     R2,-(6)         ;PUSH R2 ON STACK
026002  010346                        MOV     R3,-(6)         ;PUSH R3 ON STACK
026004  010446                        MOV     R4,-(6)         ;PUSH R4 ON STACK
026006  010546                        MOV     R5,-(6)         ;PUSH R5 ON STACK
026010  010637  026112                MOV     SP,.SAVR6       ;SAVE SP
026014  012777  026024  000072        MOV     #.POWUP,@.PUVEC ;SET UP VECTOR
026022  000000                        HALT                    ;WAIT FOR PF

026024  013706  026112  .POWUP: MOV     .SAVR6,SP       ;GET SP
026030  005001                  CLR     R1              ;WAIT LOOP FOR THE TTY
026032  005201          1$:     INC     R1              ;WAIT FOR THE INC
026034  001376                  BNE     1$              ;OF  WORD
026036  012605                  MOV     (6)+,R5         ;POP STACK INTO R5
026040  012604                  MOV     (6)+,R4         ;POP STACK INTO R4
026042  012603                  MOV     (6)+,R3         ;POP STACK INTO R3
026044  012602                  MOV     (6)+,R2         ;POP STACK INTO R2
026046  012601                  MOV     (6)+,R1         ;POP STACK INTO R1
026050  012600                  MOV     (6)+,R0         ;POP STACK INTO R0
026052  012737  025760  000024        MOV     #.POWER,@#24    ;SET UP THE POWER DOWN VECTOR
026060  012737  000340  000026        MOV     #340,@#26       ;PRIO:7
026066  104402  026072                TYPE    ,.+2            ;.ASCIZ <15><12>"POWER"
026102  000137  021754                JMP     MULSYS          ;JMP TO USER ADDRESS

026106  000000  .ILLUP: HALT                    ;THE POWER UP SEQUENCE WAS STARTED
026110  000776          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE

026112  000000  .SAVR6: 0                       ;PUT THE SP HERE
026114  000024  000026  .PUVEC: 24,26           ;POWER UP VECTOR
```

```
                        .SBTTL      SRDOCT - 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.
026120 010546 000004 000002 ,RDOCT: MOV     (SP)-(SP),      ;MOVE THE PC
026124 010546                       MOV     -(SP),-(SP)     ;MOVE THE PS
026126 010046                       MOV     R0,-(SP)        ;PUSH R0 ON STACK
026130 010146                       MOV     R1,-(SP)        ;PUSH R1 ON STACK
026132 010246                       MOV     R2,-(SP)        ;PUSH R2 ON STACK
026134 004737 027014        46:     JSR     PC,RDLIN        ;READ A LINE INTO INPUT
026140 005002                       CLR     R2              ;INIT DATA LOAD
026142 005001                       CLR     R1              ;CLEAR COUNT WORD
026144 012700 027014                MOV     #INPUT,R0       ;INIT POINTER
026150 112046               10:     MOVB    (R0)+,-(SP)     ;GET A BYTE
026152 122726 000015                CMPB    (SP)+,#15       ;IS IT A CR?
026156 001773                       BEQ     10              ;OUT IF YES
026160 122726 000060                CMPB    (SP)+,#60       ;CHECK FOR 0 OR GREATER
026164 103767                       BLO     46              ;ERROR - LESS THAN 0
026166 122726 000067                CMPB    (SP)+,#67       ;CHECK FOR 7 OR LESS
026172 103364                       BLO     20              ;
026174 103763                       BLO     46              ;ERROR - GREATER THAN 7
026176 042716 177770        20:     BIC     #177770,(SP)    ;INTO
026202 006301                       ASL     R1              ;POSITION
026204 006301                       ASL     R1              ;FIRST BIT
026206 006301                       ASL     R1              ;
026210 006301                       ASL     R1              ;SECOND BIT
026212 006301                       ASL     R1              ;
026214 006301                       ASL     R1              ;3RD BIT
026216 042701 177770                BIC     #177770,R1      ;YES HE TYPED SOMETHING
026222 050201                       BIS     R2,R1           ;OR
026224 012702 000012                MOV     #10.,R2         ;SAVE THE RESULT
026230 006202               30:     ASR     R2              ;FOR
026232 006202                       ASR     R2              ;STACK
026234 006202                       ASR     R2              ;FOR
026236 006202                       ASR     R2              ;STACK
026240 000207                       RTS     PC              ;RETURN
026242 010102               35:     MOV     R1,R2           ;
026244 000732                       BR      46              ;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.

026252  010546                  .RDLIN: MOV     R5,-(6)         ;SAVE R5
026254  012705  026374          1$:     MOV     #INPUT,R5       ;GET ADDRESS
026260  022705  026414          2$:     CMP     #INPUT+16.,R5   ;BUFFER FULL?
026264  001423                          BEQ     4$              ;YES - TYPE "?"
026266  105737  177560                  TSTB    @#177560        ;WAIT FOR
026272  100375                          BPL     .-4             ;A CHARACTER
026274  113715  177562                  MOVB    @#177562,(5)    ;GET CHARACTER
026300  142715  000200                  BICB    #200,(5)        ;GET RID OF JUNK
026304  122715  000025                  CMPB    #25,(5)         ;IS IT A ↑U
026310  001006                          BNE     5$              ;BRANCH IF NOT
026312  104402  026316                  TYPE    ,.+2            ;.ASCIZ "↑U"<15><12>
026324  000753                          BR      1$              ;START OVER
026326  122715  000177          5$:     CMPB    #177,(5)        ;IS IT A RUBOUT
026332  001005                          BNE     3$              ;SKIP IF NOT
026334                          4$:
026334  104402  026340                  TYPE    ,.+2            ;.ASCIZ "?"<15><12>
026344  000743                          BR      1$              ;ZAP THE BUFFER AND LOOP
026346  111527  000000          3$:     MOVB    (5),#0          ;SET UP FOR TYPING
026352  104402  026350                  TYPE    ,3$+2           ;ECHO IT
026356  122725  000015                  CMPB    #15,(5)+        ;CHECK FOR RETURN
026362  001336                          BNE     2$              ;LOOP IF NOT RETURN
026364  104402  000012                  TYPE    ,12             ;TYPE A LINE FEED
026370  012605                          MOV     (6)+,R5         ;RESTORE R5
026372  000002                          RTI                     ;RETURN

026374  000020                  INPUT:  .BLKB   16.             ;TTY INPUT AREA
```

```
                                .SBTTL          $TRAP - TRAP HANDLER

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

026414  011646          .TRAP:  MOV     (6),-(6)        ;GET ADDRESS OF TRAP +2
026416  162716  000002          SUB     #2,(6)          ;MAKE IT ADDRESS OF TRAP
026422  017616  000000          MOV     @(6),(6)        ;GET TRAP INSTRUCTION
026426  062716  122034          ADD     #.TRP+2-TRAP,(6);GET DATA AND MAKE IT AN OFFSET
026432  013607          .TRP:   MOV     @(6)+,PC        ;GO TO PROPER SUBROUTINE

026434  025212                  .SCOPE                  ;SCOPE   =  TRAP+0       (104400)
026436  025054                  .TYPE                   ;TYPE    =  TRAP+2       (104402)
026440  025556                  .TYPEO                  ;TYPEO   =  TRAP+4       (104404)
026442  025566                  .TYPES                  ;TYPES   =  TRAP+6       (104406)
026444  026120                  .RDOCT                  ;RDOCT   =  TRAP+10      (104410)
026446  026252                  .RDLIN                  ;RDLIN   =  TRAP+12      (104412)
026450  027016                  .CLRDK                  ;CLRDK   =  TRAP+14      (104414)
026452  027044                  .MRDMD                  ;MRDMD   =  TRAP+16      (104416)
026454  022276                  .MRCK                   ;MRCK    =  TRAP+20      (104420)
026456  022246                  .MRCLK                  ;MRCLK   =  TRAP+22      (104422)
026460  022240                  .MRINT                  ;MRINT   =  TRAP+24      (104424)
026462  022342                  .DSCK                   ;DSCK    =  TRAP+26      (104426)
026464  022324                  .MRIND                  ;MRIND   =  TRAP+30      (104430)
026466  022370                  .XBIT                   ;XBIT    =  TRAP+32      (104432)
026470  022660                  .CLKD1                  ;CLKD1   =  TRAP+34      (104434)
026472  022734                  .CLKD0                  ;CLKD0   =  TRAP+36      (104436)
026474  023664                  .CLKR1                  ;CLKR1   =  TRAP+40      (104440)
026476  024046                  .CLKR0                  ;CLKR0   =  TRAP+42      (104442)
026500  023510                  .RBIT                   ;RBIT    =  TRAP+44      (104444)
026502  023456                  .MCLK1                  ;MCLK1   =  TRAP+46      (104446)
026504  023500                  .MCLK0                  ;MCLK0   =  TRAP+50      (104450)
026506  023256                  .MCLKB                  ;MCLKB   =  TRAP+52      (104452)
026510  027136                  .GETSP                  ;GETSP   =  TRAP+54      (104454)
026512  027176                  .SPASS                  ;SPASS   =  TRAP+56      (104456)
026514  026524                  .SUSWR                  ;SUSWR   =  TRAP+60      (104460)
026516  026726                  .CNTLU                  ;CNTLU   =  TRAP+62      (104462)
026520  026644                  .KBDIN                  ;KBDIN   =  TRAP+64      (104464)
026522  026726                  .CNTLU                  ;CNTLU   =  TRAP+66      (104466)

026524  032737  000001  026642  .SUSWR: BIT     #BIT0,SWI
026532  001037                  BNE     XXX
026534  013746  000006          MOV     6,-(SP)         ;SAVE 6 ON STACK
026540  013746  000004          MOV     4,-(SP)         ;SAVE 4 ON STACK
026544  012737  026564  000004  MOV     #1$,4           ;SET UP TRAP ADDRESS
026552  022777  177777  152246  CMP     #-1,@SWR        ;TEST 177570
026560  001402                  BEQ     2$              ;FAKE OUT
026562  000407                  BR      3$              ;HARDWARE AVAILABLE
026564  022626          1$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
026566  012737  000176  001026  2$:     MOV     #SWREG,SWR      ;SET UP SOFTWARE REGISTERS
026574  012737  000174  001030          MOV     #DISPREG,DISPLAY
026602  022737  000176  001026  3$:     CMP     #SWREG,SWR      ;1ST TIME THRU?
026610  001004                  BNE     4$              ;NO CHANGE STILL 177570
```

```
5198  026612  005737  000042          TST     42              ;ANY XXDP OR ACT
5199  026616  001001                  BNE     4$              ;SWR=000000
5200  026620  104466                  CNTLU                   ;GET INITIAL SETTINGS
5201  026622  012637  000004  4$:     MOV     (SP)+,4         ;REPLACE 4 FROM STACK
5202  026626  012637  000006          MOV     (SP)+,6         ;REPLACE 6 FROM STACK
5203  026632  052737  000001  026642  XXX:    BIS     #BIT0,SWI       ;SET THE BEENHEREBIT
5204  026640  000002                  RTI                     ;ALL DONE
5205
5206  026642  000000          SWI:    0
5207
5208
5209
5210  026644  005737  000042  .KBDIN: TST     42              ;GOT XXDP OR ACT
5211  026650  001057                  BNE OKT                 ;YES,GET OUT
5212  026652  022737  000176  001026  CMP     #SWREG,SWR      ;GOT SWITCH-LESS MACHINE?
5213  026660  001053                  BNE     OKT             ;NO GET OUT
5214  026662  105777  152132          TSTB    @TKS            ;HAVE A CHARACTER
5215  026666  100050                  BPL     OKT             ;NO GET OUT
5216  026670  017737  152126  027012  MOV     @TKB,.MSG
5217  026676  042737  177600  027012  BIC     #177600,.MSG    ;STRIP OFF GARBAGE
5218  026704  122737  000007  027012  CMPB    #7,.MSG         ;DO WE HAVE A ↑G
5219  026712  001036                  BNE OKT                 ;NO,GET OUT
5220  026714  104402  026720          TYPE    ,.+2            ;.ASCIZ <15><12>"↑G"
5221  026726                  .CNTLU:
5222  026726  104402  026732          TYPE    ,.+2            ;.ASCIZ <15><12>"SWR= "
5223  026742  013746  000176          MOV     $WREG,-(6)      ;PUT SWREG ON STACK
5224  026746  104404                  TYPE0                   ;TYPE STACK IN OCTAL
5225  026750  104402  026754          TYPE    ,.+2            ;.ASCIZ "    NEW= "
5226  026766  104410                  RDOCT
5227  026770  012637  027012          MOV     (SP)+,.MSG      ;GET NEW VALUE OFF STACK
5228  026774  005737  027014          TST     CTN             ;DID HE TYPE <CR> OF 000000?
5229  027000  001403                  BEQ     OKT             ;DONT CHANGE IF <CR>
5230  027002  013737  027012  000176  MOV     .MSG,SWREG      ;CHANGE VALUE OF SWREG
5231  027010  000002          OKT:    RTI                     ;ALL DONE-EXIT
5232
5233  027012  000000          .MSG:   0
5234  027014  000000          CTN:    0
5235
```

```
5236                                            ;CLEAR ALL DISK REGISTERS
5237  027016  012777  000040  152056  .CLRDK:   MOV     #40,@RSCS2      ;CLEAR ALL DSK REG
5238  027024  013777  001162  152050           MOV     UNNUM,@RSCS2    ;GET UNIT NUMBER
5239  027032  005037  001176                   CLR     MCCNT           ;CLEAR MAINT CLOCK COUNT
5240  027036  005037  001200                   CLR     MCCNT+2
5241  027042  000002                           RTI
5242
5243  027044  012777  000001  152052  .MRDMD:   MOV     #1,@RSMR        ;PUT DRIVE INTO MAINT MODE
5244  027052  000002                           RTI
5245
5246  027054  005037  001216          WAITRY:   CLR     WORK            ;CLEAR COUNTER
5247  027060  105777  152014          1$:       TSTB    @RSCS1          ;TEST READY
5248  027064  100406                           BMI     2$              ;OK CONT
5249  027066  005237  001216                   INC     WORK            ;UPDATE COUNTER
5250  027072  005737  001216                   TST     WORK            ;DONE YET?
5251  027076  001403                           BEQ     3$              ;READY DID NOT COME UP
5252  027100  000767                           BR      1$              ;CONTINUE WAITING
5253  027102  062716  000002          2$:       ADD     #2,(SP)         ;UPDATE RETURN PC
5254  027106  000207                  3$:       RTS     PC              ;RETURN
5255
5256                                            ;ROUTINE TO SHIFT COMPLETE DATA TABLE ONE BIT
5257                                            ;TO THE LEFT.  CARRIES BIT 15 IF ONE RVD TO BIT 0 OF THE NEXT WORD
5258
5259  027110  012702  030114          MDATA:    MOV     #INBUF,R2       ;GET LEFT ADDRESS OF
5260  027114  062702  000442                   ADD     #442,R2         ;DATA TABLE
5261  027120  012703  000220                   MOV     #220,R3         ;SETUP COUNTER FOR 200 WORDS
5262  027124  000241                           CLC                     ;CLEAR CARRY
5263  027126  006142                  1$:       ROL     -(R2)           ;SHIFT DATA PATTERN
5264  027130  005303                           DEC     R3              ;DO ALL
5265  027132  001375                           BNE     1$              ;WORDS
5266  027134  000207                           RTS     PC
5267  027136  012737  001001  001204  .GETSP:   MOV     #1001,REPT      ;SETUP COUNTER
5268  027144  104430                           MRIND                   ;SEND INDEX PULSE TO MR REG
5269  027146  104422                  1$:       MRCLK                   ;CLOCK MR
5270  027150  005337  001204                   DEC     REPT            ;TO REACH
5271  027154  001374                           BNE     1$              ;SECTOR PULSE
5272  027156  032777  000400  151740          BIT     #400,@RSMR      ;DID SECTOR PULSE SET?????
5273  027164  001401                           BEQ     2$              ;YES
5274  027166  000002                           RTI                     ;NO REPORT ERROR
5275  027170  062716  000002          2$:       ADD     #2,(SP)         ;UPDATE RETURN ADDR
5276  027174  000002                           RTI
5277
5278  027176  104422                  .SPASS:   MRCLK                   ;CLOCK PAST SECTOR PULSE
5279  027200  104422                           MRCLK
5280  027202  005037  001176                   CLR     MCCNT           ;RESET MAINT CLOCK COUNTERS
5281  027206  005037  001200                   CLR     MCCNT+2
5282  027212  000002                           RTI
```

```
                                        ;ERROR TYPTXTOUT ROUTINE
                                        
027214  005737  025544          RSREG:  TST     .HLTCT          ;SHOULD WE TYPTXT GOOD AND BAD
027220  001022                          BNE     8$              ;NO
027222  104402  027226                  TYPE    ,.+2            ;.ASCIZ " BAD="
027234  010046                          MOV     @#BAD,-(6)      ;PUT BAD ON STACK
027236  104404                          TYPEO                   ;TYPE STACK IN OCTAL
027240  104402  027244                  TYPE    ,.+2            ;.ASCIZ " GOOD="
027254  010146                          MOV     @#GOOD,-(6)     ;PUT GOOD ON STACK
027256  104404                          TYPEO                   ;TYPE STACK IN OCTAL
027260  000402                          BR      8$              ;TYPEOUT REGISTERS
027262  000137  027720                  JMP     @#PTDONE        ;GET OUT
027266                          8$:
027266  104402  027272                  TYPE    ,.+2            ;.ASCIZ " CS1="
027300  017746  151574                  MOV     @#RSCS1,-(6)    ;PUT @#RSCS1 ON STACK
027304  104404                          TYPEO                   ;TYPE STACK IN OCTAL
027306                          1$:
027306  104402  027312                  TYPE    ,.+2            ;.ASCIZ " ER="
027320  017746  151570                  MOV     @#RSER,-(6)     ;PUT @#RSER ON STACK
027324  104404                          TYPEO                   ;TYPE STACK IN OCTAL
027326                          2$:
027326  104402  027332                  TYPE    ,.+2            ;.ASCIZ " CS2="
027340  017746  151536                  MOV     @#RSCS2,-(6)    ;PUT @#RSCS2 ON STACK
027344  104404                          TYPEO                   ;TYPE STACK IN OCTAL
027346  032737  000200  025544          BIT     #200,.HLTCT     ;TYPTXT SECOND SET ?
027354  001076                          BNE     SEEC            ;YES
027356  032737  000100  025544          BIT     #AS,.HLTCT      ;TYPTXT ER ?
027364  001410                          BEQ     3$              ;NO
027366  104402  027372                  TYPE    ,.+2            ;.ASCIZ " AS="
027400  017746  151512                  MOV     @#RSAS,-(6)     ;PUT @#RSAS ON STACK
027404  104404                          TYPEO                   ;TYPE STACK IN OCTAL
027406  032737  000020  025544  3$:     BIT     #BA,.HLTCT      ;TYPTXT BUS ASSRESS
027414  001410                          BEQ     4$              ;NO
027416  104402  027422                  TYPE    ,.+2            ;.ASCIZ " BA="
027430  017746  151452                  MOV     @#RSBA,-(6)     ;PUT @#RSBA ON STACK
027434  104404                          TYPEO                   ;TYPE STACK IN OCTAL
027436  032737  000004  025544  4$:     BIT     #DA,.HLTCT      ;TYPTXT DA ?
027444  001410                          BEQ     5$              ;NO
027446  104402  027452                  TYPE    ,.+2            ;.ASCIZ " DA="
027460  017746  151424                  MOV     @#RSDA,-(6)     ;PUT @#RSDA ON STACK
027464  104404                          TYPEO                   ;TYPE STACK IN OCTAL
027466  032737  000010  025544  5$:     BIT     #WC,.HLTCT      ;TYPTXT WC?
027474  001410                          BEQ     6$              ;NO
027476  104402  027502                  TYPE    ,.+2            ;.ASCIZ " WC="
027510  017746  151370                  MOV     @#RSWC,-(6)     ;PUT @#RSWC ON STACK
027514  104404                          TYPEO                   ;TYPE STACK IN OCTAL
```

```
5329  027516  032737  000040  025544  6$:     BIT     #DS,.HLTCT      ;DRIVE STATUS
5330  027524  001475                          BEQ     PTDONE          ;NO
5331  027526  104402  027532                  TYPE    .+2             ;.ASCIZ " DS="
5332  027540  017746  151346                  MOV     @RSDS,-(6)      ;PUT @RSDS ON STACK
5333  027544  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5334  027546  000137  027720                  JMP     PTDONE          ;GET OUT
5335  027552  042737  000200  025544  SEEC:   BIC     #200,.HLTCT     ;CLEAR COMMON BIT
5336  027560  032737  000240  025544          BIT     #DT,.HLTCT      ;TYPTXT DRIVE TYPE?
5337  027566  001410                          BEQ     9$              ;NO
5338  027570  104402  027574                  TYPE    .+2             ;.ASCIZ " DT="
5339  027602  017746  151320                  MOV     @RSDT,-(6)      ;PUT @RSDT ON STACK
5340  027606  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5341  027610  032737  000210  025544  9$:     BIT     #DB,.HLTCT      ;TYPTXT DATA BUFFER
5342  027616  001410                          BEQ     10$             ;NO
5343  027620  104402  027624                  TYPE    .+2             ;.ASCIZ " DB="
5344  027632  017746  151264                  MOV     @RSDB,-(6)      ;PUT @RSDB ON STACK
5345  027636  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5346  027640  032737  000220  025544  10$:    BIT     #MR,.HLTCT      ;TYPTXT MN?
5347  027646  001410                          BEQ     11$             ;NO
5348  027650  104402  027654                  TYPE    .+2             ;.ASCIZ " MR="
5349  027662  017746  151236                  MOV     @RSMR,-(6)      ;PUT @RSMR ON STACK
5350  027666  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5351  027670  032737  000204  025544  11$:    BIT     #LA,.HLTCT      ;TYPTXT LA?
5352  027676  001410                          BEQ     PTDONE          ;NO
5353  027700  104402  027704                  TYPE    .+2             ;.ASCIZ " LA="
5354  027712  017746  151202                  MOV     @RSLA,-(6)      ;PUT @RSLA ON STACK
5355  027716  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5356  027720  052737  100000  001170  PTDONE: BIS     #BIT15,ONCEE    ;SET FORND ERROR FLAG
5357  027726  032737  000040  001170          BIT     #BIT5,ONCEE
5358  027734  001466                          BEQ     1$
5359  027736  104402  027742                  TYPE    .+2             ;.ASCIZ <15><12>"MAINT CLOCK COUNT "
5360  027770  013737  001176  001230          MOV     MCCNT,WORK4     ;GET MAINT CLOCK COUNT
5361  027776  013737  001200  001224          MOV     MCCNT+2,WORK2   ;CAL NUMBERS FOR DOUBLE PRECISION
5362  030004  006137  001224                  ROL     WORK2
5363  030010  006137  001230                  ROL     WORK4
5364  030014  000241                          CLC
5365  030016  013746  001230                  MOV     WORK4,-(6)      ;PUT WORK4 ON STACK
5366  030022  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
5367  030024  012737  000005  001232          MOV     #5,WORK5
5368  030032  005037  001234          2$:     CLR     WORK6
5369  030036  006137  001224                  ROL     WORK2
5370  030042  006137  001234                  ROL     WORK6
5371  030046  006137  001224                  ROL     WORK2
5372  030052  006137  001234                  ROL     WORK6
5373  030056  006137  001224                  ROL     WORK2
5374  030062  006137  001234                  ROL     WORK6
5375  030066  013746  001234                  MOV     WORK6,-(6)      ;PUT WORK6 ON STACK
5376  030072  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
5377  030074  005337  001232                  DEC     WORK5
5378  030100  001354                          BNE     2$
5379  030102  104402  030106                  TYPE    .+2             ;.ASCIZ <15><12>
5380  030112  000207          1$:     RTS     PC
5381  030114  000300          INBUF:  .BLKW   300
5382  030714  000300          OUTBUF: .BLKW   300
```

```
                                        ;THIS ROUTINE IS FOR PROGRAMMERS ONLY !!!!!!!!!!!!!!THIS ROUTINE IS USED TO "DETERMINE" A
                                        ;SO THAT A 1 CAN BE ROTATED THROUGH THE CRC REGISTER BY ROTATING THE DATA PATTERN
031514  012737  000040  001144  CRCAL:  MOV     #40,FLAG2
031522  012706  000500                  MOV     #500,SP
031526  005037  001226                  CLR     WORK3
031532  012702  030114                  MOV     #INBUF,R2
031536  012701  000221                  MOV     #145.,R1
031542  005022                  1$:     CLR     (R2)+           ;CLEAR DATA BUFFER
031544  005301                          DEC     R1
031546  001375                          BNE     1$
031550  012737  000401  001226          MOV     #401,WORK3      ;START WITH A NUMBER OF 401
031556  012702  030114          3$:     MOV     #INBUF,R2
031562  062702  000100                  ADD     #100,R2
031566  062737  000003  001226          ADD     #3,WORK3
031574  013712  001226                  MOV     WORK3,(R2)      ;PUT NUMBER INTO BUFFER
031600  012701  001001          2$:     MOV     #513.,R1        ;513=32 WORDS X 16 BITS
031604  005301                  6$:     DEC     R1
031606  001763                          BEQ     3$
031610  012700  000040                  MOV     #40,R0
031614  012702  030114                  MOV     #INBUF,R2
031620  062702  000102                  ADD     #102,R2
031624  000241                          CLC
031626  006142                  5$:     ROL     -(R2)
031630  005300                          DEC     R0
031632  001375                          BNE     5$
031634  004737  024430                  JSR     PC,GENCRC
031640  022737  000001  001216          CMP     #1,WORK
031646  001013                          BNE     4$
031650  104402  031654                  TYPE    .+2             ;.ASCIZ <15><12>"CRC= "
031664  013746  001216                  MOV     WORK,-(6)       ;PUT WORK ON STACK
031670  104404                          TYPEO                   ;TYPE STACK IN OCTAL
031672  004737  031736                  JSR     PC,TABTYP
031676  022737  000002  001216  4$:     CMP     #2,WORK
031704  001337                          BNE     6$
031706  104402  031712                  TYPE    .+2             ;.ASCIZ <15><12>"CRC= "
031722  013746  001216                  MOV     WORK,-(6)       ;PUT WORK ON STACK
031726  104404                          TYPEO                   ;TYPE STACK IN OCTAL
031730  004737  031736                  JSR     PC,TABTYP
031734  000723                          BR      6$
031736  012702  030114          TABTYP: MOV     #INBUF,R2
031742  012705  000220                  MOV     #220,R5
031746  012737  000004  001204  2$:     MOV     #4,REPT
031754                          1$:
031754  012246                          MOV     (R2)+,-(6)      ;PUT (R2)+ ON STACK
031756  104404                          TYPEO                   ;TYPE STACK IN OCTAL
031760  104402  000040                  TYPE    ,40
031764  005305                          DEC     R5
031766  001410                          BEQ     3$
031770  005337  001204                  DEC     REPT
031774  001367                          BNE     1$
031776  104402  032002                  TYPE    .+2             ;.ASCIZ <15><12>
032006  000757                          BR      2$
032010  000207                  3$:     RTS     PC

        000001                          .END
```

```
AOECK   021054          4251#
AS    = 000100           899#  1177  2248  2252  2255  2355  2359  2362  2618  3733  5309
ATA   = 100000           920#
BA    = 000020           897#  1147  1206  1212  1294  1298  1302  1324  1510  1514  5314
BAD   =X000000           843#  1036* 1271* 1310* 1311  1348* 1349  1385* 1386  1487* 1488  1524* 1525*
                        1528   1538* 1540  1554* 1555  1561* 1562  1566* 1567  1591* 1582  1589* 1590
                        1611*  1613  1635* 1636  1643* 1644  1832* 1834  1844* 1846  1909* 1911  1920*
                        1922   1932* 1934  1944* 1946  1957* 1978  1998* 2000  2009* 2011  2016* 2018
                        2054*  2056  2067* 2069  2076* 2078  2084* 2086  2167* 2168  2179* 2181  2186*
                        2189   2194* 2196  2390* 2392  2441* 2443  2495* 2497  2543* 2545  2604* 2607
                        2630*  3092* 3097  3103  3104  3106* 3109* 3111  3117  3740* 3743  3760* 3769*
                        3772   3855* 3857  4428* 4431  4440* 4443  4520* 4521  4538* 4539  4697* 4698
                        4722*  4723  5288
BAI   = 000010           923#  4342
BEGIN   001240           806    988#
BEGIN1  000232           802    808#
BEGIN2  000226           806#   809
BELL  = 000007           817#  4369  4959
BITBA   004674          1506#
BITCS2  004560          1482#
BITST   004506          1469#
BITWC   004624          1494#
BIT0  = 000001           826#  1092  1111  3731  4359  4817  4820  4828  4841  5185  5203
BIT1  = 000002           827#  2040  2062  2066  2071  2073  2081  2083  2088  2091  2882  4826
BIT10 = 002000           836#  1960  1976  1993  4475  4496  4498  4753  4761  4774  4777
BIT11 = 004000           837#  4289  4646  4653  4658  4660  4691  4716  4748
BIT12 = 010000           838#  2592  2635  2638  3099  4515  4532  4534
BIT13 = 020000           839#  1054  1069  1075  1096  3112  4766  4771
BIT14 = 040000           795    840#  4839
BIT15 = 100000           841#  1057  1077  1084  1539  4844  4847  5356
BIT2  = 000004           828#  2118  2135  3796  3804  3808  3811  3814  3817  3820  4391  4393  4679
                        4706   4732  4831  4834
BIT3  = 000010           829#  2123  2139  2173  2201  2203  2205  4302  4317  4703
BIT4  = 000020           830#   997  1009  1086  3333  3364  3530  3559  3895  4384
BIT5  = 000040           831#  2039  2219  2268  2280  2325  2373  2478  2703  2896  2957  3090  3131
                        3181   3326  3383  3523  3577  3789  3890  3984  4650  4785  4798  4806  5357
BIT6  = 000100           805    808   832#  1001  2790  4071  4454  4458  4681  4708
BIT7  = 000200           833#  2801  2808  2813  4082  4089  4095  4450  4456  4739
BIT8  = 000400           834#  2803  2812  3018  3020  3242  3244  3443  3445  3637  3639  4084  4094
                        4460   4464  4467  4648  4683  4692  4710  4717  4741
BIT9  = 001000           835#  1823  1825  1828  1967  1969  1971  4466  4471  4482  4499
CALRTB  024202          4672   4729#
CHCKDV  002376          1089   1092#
CHG     022122          2184   2192   2199  4398#
CLKCNT  001210           945#  4459#  4647# 4738#  4743  4746  4749  4751  4757  4759  4763  4772  4775
CLKD0 = 104436          2796   4077   5170#
CLKD1 = 104434          2792   4073   5169#
CLKR0 = 104442          3013   3237   3438  3632  5172#
CLKR1 = 104440          3010   3234   3435  3629  5171#
CLROK = 104414          1236   1469   1515  1523  1548  1573  1619  1656  1685  1863  2038  2075  2117
                        2157   2218   2267  2372  2408  2424  2459  2477  2510  2527  2557  2578  2654
                        2895   3130   3325  3522  3720  3768  3821  3839  3870  3889  3912  3934  4187
                        4263   4293   4315  4337  5161#
CNTLU = 104466          5180#  5182#  5200
CRCAL   031514          5385#
CRCTAB  023116          4547   4559#
```

```
CRCTYP  023006          3702    4547#
CRCWD   024372          4742    4771#
CS1     = 000001         893#   1142    1201    1241    1245    1249    1257    1474    1478    1533    1537    1664    1669
CS2     = 000200         900#   1133    1197    1267    1277    1281    1286    1530    1542    1559    1584    1595    1607
CTN     027C14          5081#   5098#   5228    5234#
DA      = 000004         895#   1153    1369    1373    1377    1398    2407    2458    2613    2629    2875    2881    3078
                        3084    3303    3309    3500    3694    3805    3812    3818    3905    3909    3915    3919    4166
                        4169    4175    5319
DAO     = 001000         921#
DB      = 000210         902#   5341
DCK     = 100000         922#
DISPLA  001030           858#   4937#   4941#   5195#
DISPRE  000174           798#   5195
DLT     = 100000         915#   1605
DONE    021672          1106    4363#
DONEE   002552          1087    1093    1105#
DRY     = 000200         916#
DS      = 000040         898#   1137    2064    2130    2134    2145    2149    2248    2252    2255    2355    2359    2362
                        2402    2411    2415    2453    2462    2466    2507    2513    2517    2554    2560    2564    2622
                        3078    3303    3500    3694    3700    3733    3751    3805    3812    3818    3824    3864    3868
                        3874    3878    3900    3905    3915    3919    4158    4172    4258    4261    4266    4269    4313
                        4328    4351    4355    5329
DSCK    = 104426        2230    2236    2243    2305    2311    2350    2380    2397    2431    2448    2485    2502    2534
                        2549    4222    5166#
DT      = 000240         904#   5336
DVNUM   002036          1048    1051#
ER      = 000002         894#   1159    1218    1406    1410    1417    1423    1430    3751
ERR     = 040000         919#
ERRORS  001002           848#   4960#   4980
E1910   023440          4587    4622#
E1915   023447          4589    4624#
E192    023424          4583    4620#
E197    023432          4585    4621#
E2410   023377          4577    4614#
E2412   023406          4579    4616#
E2415   023415          4581    4618#
E242    023355          4571    4608#
E245    023363          4573    4610#
E247    023371          4575    4612#
E3010   023330          4565    4602#
E3012   023337          4567    4604#
E3015   023346          4569    4606#
E302    023306          4559    4599#
E305    023314          4561    4600#
E307    023322          4563    4601#
FILCHR  001014           852#   4885    4886
FLAG2   001144           927#    999#   2653#   2882    2904#   3139#   3324#   3333    3364#   3521#   3530    3559#   3933#
                        4292#   4450    4456#   4464    4466#   4467#   4471    4475    4479#   4482    4496    4498#   4499#
                        4646#   4650    4653#   4658    4660#   4679    4706    4785    4798    4806    5385#
FLAG3   001146           805#    808#    928#    997#   1001    1009#   1086    4384
FLOTBA  003616          1307#
FLOTDA  004116          1382#
FLOTWC  003756          1345#
GENCRC  024430          2814    3022    3246    4101    4784#   5408
GETSP   = 104454        2689    2942    3167    3368    3563    3970    5177#
GOOD    =%000001         842#   1037#   1272#   1307#   1309    1311    1314#   1345#   1347    1349    1352#   1382#   1384
```

```
                    1386    1389*   1485*   1486*   1488    1526*   1527*   1528    1539*   1540    1552*   1553*   1555
                    1560*   1562    1565*   1567    1579*   1580*   1582    1585*   1586*   1587    1590    1612*   1613
                    1621*   1622    1623*   1626*   1628    1629*   1634*   1636    1639*   1642*   1644    1648*   1833*
                    1834    1845*   1846    1910*   1911    1921*   1922    1933*   1934    1945*   1946    1977*   1978
                    1999*   2000    2010*   2011    2017*   2018    2055*   2056    2068*   2069    2077*   2078    2085*
                    2086    2165*   2166*   2168    2180*   2181    2187*   2188*   2189    2195*   2196    2391*   2392
                    2442*   2443    2496*   2497    2544*   2545    2589*   2590    2594*   2599    2600    2605*   2607
                    2631*   3091*   3097    3103    3105    3107*   3110*   3117    3741*   3742*   3743    3761*   3770*
                    3771*   3772    3856*   3857    4429*   4431    4441*   4443    4517*   4518*   4521    4535*   4536*
                    4539    4678*   4681*   4682*   4687*   4691*   4696*   4698    4705*   4708*   4709*   4712*   4716*
                    4721*   4723    4748*   4753*   4761*   4766*   4771*   4774*   4777*   5291
HLT     = 104000     814#   1038    1133    1137    1142    1147    1153    1159    1165    1171    1177    1197    1201
                    1206    1212    1218    1223    1228    1241    1245    1249    1257    1267    1273    1277    1281
                    1286    1294    1298    1302    1313    1324    1332    1336    1340    1351    1361    1369    1373
                    1377    1388    1398    1406    1410    1417    1423    1430    1440    1447    1455    1463    1474
                    1478    1490    1498    1502    1510    1514    1530    1533    1537    1542    1559    1564    1569
                    1584    1592    1595    1607    1615    1638    1646    1664    1669    1696    1702    1714    1718
                    1727    1731    1734    1743    1747    1756    1760    1770    1774    1780    1787    1791    1802
                    1816    1820    1836    1848    1877    1885    1889    1892    1901    1905    1913    1924    1936
                    1948    1980    2002    2013    2020    2023    2063    2064    2072    2082    2090    2130    2134
                    2145    2149    2171    2185    2193    2200    2232    2238    2245    2248    2252    2255    2277
                    2284    2288    2295    2299    2307    2313    2320    2329    2333    2342    2346    2352    2355
                    2359    2362    2382    2396    2399    2402    2407    2411    2415    2433    2447    2450    2453
                    2458    2462    2466    2487    2501    2504    2507    2513    2517    2536    2548    2551    2554
                    2560    2564    2609    2613    2618    2622    2629    2632    2692    2698    2707    2711    2720
                    2724    2732    2736    2744    2748    2752    2758    2762    2771    2778    2782    2795    2800
                    2831    2838    2842    2850    2854    2858    2865    2869    2875    2978    2881    2945    2952
                    2961    2965    2974    2978    2985    2989    2999    3011    3014    3035    3039    3043    3047
                    3057    3061    3071    3078    3081    3084    3108    3170    3176    3185    3189    3198    3202
                    3210    3214    3223    3235    3238    3259    3263    3267    3271    3281    3285    3295    3303
                    3306    3309    3371    3378    3387    3391    3400    3404    3411    3415    3424    3436    3439
                    3456    3460    3464    3468    3478    3482    3492    3500    3503    3566    3572    3591    3585
                    3594    3598    3605    3609    3618    3630    3633    3650    3654    3658    3662    3672    3676
                    3686    3694    3697    3700    3733    3745    3751    3755    3762    3774    3805    3812    3818
                    3824    3860    3864    3868    3874    3878    3900    3905    3909    3915    3919    3973    3979
                    3988    3992    4001    4005    4013    4017    4025    4029    4033    4039    4043    4052    4056
                    4063    4076    4081    4116    4124    4128    4137    4141    4148    4152    4156    4158    4166
                    4169    4172    4175    4224    4232    4239    4243    4255    4258    4261    4266    4269    4310
                    4313    4325    4328    4351    4355
HLTADR  001012       851#   4964#   4965*   4966    4967    4981
IADONE  017176      3799    3826#
ICNT    001000       847#    995*   4383*   4915    4930    4932    4934*   4935*   4937    4940*   4941    4947    4976*
IE      = 000100     924#
ILFDON  007566      2108    2110#
INBIT   001212       946#   4810*   4812*   4814*   4818    4821    4829*   4832    4842*   4845    4861    4862*   4863*
INBUF   030114      2668    2686    2783    2816    2912    3003    3024    3091    3147    3164    3227    3248    3346
                    3428    3446    3541    3622    3640    3949    3967    4064    4103    4301*   4303    4318    4343
                    4344*   4788    5259    5381#   5388    5394    5402    5422
INFTST  021660      4291    4359#
INPUT   026374      1007    5082    5117    5118    5141#
INT     005444      1656#
INTDON  005534      1665    1670#
INTMR   020560      3965    4164#
INTMR1  020630      4159    4176#
IR      = 000100     910#
KBDIN   = 104464    4912    4956    5181#
```

```
LA     = 000204          901#  1702  1734  1802  1892  5351
LAD      001010          850#   996*  3791*  4382*  4936*  4942  4944  4947
LAFLAG   001236          956#   998*  1067  1105*
LBT    = 002000          918#
LSTEV    001150          929#
LSTOD    001152          930#  4452*  4457*  4470*  4512
LSTWD    024324         4740  4757#
MCCNT    001176          941#  1866*  4229*  4424*  4425*  4596*  4597*  4630*  4631*  4674*  4675*  5239*  5240*
                        5280*  5281*  5360  5361
MCLKB  = 104452         1810  1841  5176#
MCLKO  = 104450         1715  1728  1744  1757  1771  1784  1813  1886  1902  2285  2296  2330  2343
                        2708  2721  2733  2745  2755  2768  2779  2835  2847  2855  2866  2962  2975
                        2986  3036  3044  3058  3075  3186  3199  3211  3260  3268  3282  3300  3388
                        3401  3412  3457  3465  3479  3497  3582  3595  3606  3651  3659  3673  3691
                        3989  4002  4014  4026  4036  4049  4060  4121  4134  4145  4153  4240  4529
                        5175#
MCLK1  = 104446         1711  1724  1740  1753  1767  1777  1788  1817  1882  1898  2242  2281  2292
                        2326  2339  2349  2704  2717  2729  2741  2749  2759  2775  2827  2839  2851
                        2862  2958  2971  2982  3032  3040  3054  3068  3182  3195  3207  3256  3264
                        3278  3292  3384  3397  3408  3453  3461  3475  3489  3578  3591  3602  3647
                        3655  3669  3683  3985  3998  4010  4022  4030  4040  4053  4113  4125  4138
                        4149  4236  4507  5174#
MDATA    027110         3506  3706  5259#
MPRO   = 172100          939#
MR     = 000220          903#  1165  1223  1440  1447  1455  1463  1702  1734  1802  2692  2945  3170
                        3371  3566  3973  5346
MRADE    020634         4187#
MRCAL    024224         4519  4537  4738#
MRCILF   007264         2075#
MRCK   = 104420         1688  1694  1712  1716  1725  1729  1741  1745  1754  1758  1768  1772  1778
                        1785  1789  1814  1818  1869  1875  1883  1887  1899  1903  2021  2046  2120
                        2159  2221  2270  2275  2282  2286  2293  2297  2318  2327  2331  2340  2344
                        2375  2426  2480  2529  2580  2657  2696  2705  2709  2718  2722  2730  2734
                        2742  2746  2750  2756  2760  2769  2776  2780  2829  2836  2840  2848  2852
                        2856  2863  2867  2899  2950  2959  2963  2972  2976  2983  2987  2997  3033
                        3037  3041  3045  3055  3059  3069  3134  3174  3183  3187  3196  3200  3208
                        3212  3221  3257  3261  3265  3269  3279  3283  3293  3329  3376  3385  3389
                        3398  3402  3409  3413  3422  3454  3458  3462  3466  3476  3480  3490  3526
                        3570  3579  3583  3592  3596  3603  3607  3616  3648  3652  3656  3660  3670
                        3674  3684  3722  3793  3844  3892  3939  3977  3986  3990  3999  4003  4011
                        4015  4023  4027  4031  4037  4041  4050  4054  4061  4114  4122  4126  4135
                        4139  4146  4150  4154  4190  4230  4237  4241  4253  5163#
MRCLK  = 104422         1919  1930  1943  1953  1997  2008  2235  2301  2310  4221  4251  4252  4418
                        4419  5164#  5269  5278  5279
MRCRC    015044         3325#  3507
MRCRC1   015250         3384#  3393
MRCRC2   015324         3408#  3417
MRCRC3   015374         3427#
MRCRC4   015554         3473#
MRCRC5   015620         3497#
MRDCK    015672         3522#  3707
MRDCK1   016076         3578#  3587
MRDCK2   016152         3602#  3611
MRDCK3   016222         3621#
MRDCK4   016402         3667#
MRDCK5   016446         3691#
```

```
MRDMD = 104416          2045    2119    2158    2220    2269    2374    2425    2479    2528    2579    3721    3792    3843
                        3891    5162#
MRDSEL  012000          2578#
MRD3    013560          3002#
MRD4    013760          3053#
MRD5    014024          3075#
MRD6    014062          3090#
MREX    017642          3934#
MREX1   020030          3985#   3994
MREX2   020104          4010#   4019
MREX3   020170          4036#   4045
MRIFT   016562          3720#
MRILF   007046          2038#
MRIND = 104430          1687    1692    1868    1973    2014    2052    2162    2224    2274    2317    2656    2695    2898
                        2949    3133    3173    3328    3375    3525    3569    3850    3851    3852    3938    3976    4189
                        4206    4228    5167#   5268
MRINT = 104424          1690    1871    2048    2122    2161    2223    2272    2377    2428    2482    2531    2582    2659
                        2901    3136    3331    3528    3724    3795    3846    3894    3941    4192    5165#
MRLF1   007072          2041    2045#   2109
MRLF2   007104          2053#
MRLF3   007504          2094#   2096    2098    2100    2102    2104    2106
MROP    007570          2117#
MROPER  007760          2157#   2204
MROPI   017200          3839#
MROPIA  017372          3867    3870#
MRPAR   017514          3889#
MRRD    013244          2895#
MRRD1   013434          2958#   2967
MRRD2   013510          2982#   2991
MRRT1   010560          2281#   2290
MRSRC   010514          2267#
MRSRCH  010332          2218#
MRTIME  005536          1685#
MRTIM1  005604          1711#   1720
MRT2    005662          1733    1738#
MRT2A   005672          1740#   1749
MRT2B   005726          1753#   1762
MRT2C   005762          1767#   1776
MRT3    006100          1809#   1839
MRT4    006272          1863#
MRT4A   006344          1882#   1894
MRT4B   006414          1902#   1990
MRT4C   006446          1919#   1926
MRT4D   006520          1942#   1982
MRT4E   006526          1943#   1950    1994
MRT4F   006746          1955    1996#
MRT4G   007000          2008#
MRVR    021140          4289#
MRVRR   021444          4327    4334#
MRVR1   021274          4312    4315#
MRVR2   021450          2884    4337#
MRWCK   014232          3130#
MRWCK1  014406          3182#   3191
MRWCK2  014462          3207#   3216
MRWCK3  014532          3226#
MRWCK4  014732          3277#
```

```
MRWCKS  014776          3300#
MRWRT   012320          2654#  4334
MRWRT1  012460          2704#  2713
MRWRT2  012534          2729#  2738
MRWRT3  012620          2755#  2764
MRWR1   010730          2326#  2335
MULSYS  021754          4374   4380#  5062
MULTII  002006          1003   1008   1044#  4386
N     = 000070           813#  1114   1118#  1178   1182#  1229   1233#  1250   1251#  1259   1263#  1288   1289#
                        1304   1305#  1316   1317#  1325   1326#  1341   1342#  1355   1356#  1362   1363#  1378
                        1379#  1392   1393#  1399   1400#  1412   1413#  1418   1419#  1424   1425#  1431   1432#
                        1441   1442#  1448   1449#  1456   1457#  1464   1468#  1480   1481#  1492   1493#  1504
                        1505#  1516   1520#  1543   1547#  1570   1571#  1616   1617#  1652   1656#  1671   1675#
                        1849   1853#  2024   2028#  2111   2115#  2151   2155#  2206   2210#  2256   2260#  2363
                        2367#  2416   2420#  2468   2472#  2518   2522#  2565   2569#  2640   2644#  2885   2889#
                        3120   3124#  3310   3314#  3509   3513#  3709   3713#  3777   3781#  3827   3831#  3879
                        3883#  3921   3925#  4177   4181#  4282   4286#
NED   = 010000           913#  2620
NEDDON  012220          2596   2635#
NNDD    012310          2628   2633   2636   2639#
NOPERR  022026          2129   2133   2144   2148   4391#
NOWEV   001154           931#
NOWGO   002564          1041   1085   1104   1111#  4387
NOWOD   001156           932#  4462*  4470   4481*  4486*  4493*  4495*  4509   4530   4655*  4662*  4664*  4668*
                        4670*  4730
OKT     027010          5211   5213   5215   5219   5229   5231#
ONCEE   001170           937#  1000*  1057*  1077   1084*  1092   1111*  1686*  1823   1825*  1828*  1864*  1865*
                        1960   1967   1969*  1971*  1976*  1993*  2039*  2040   2062*  2066*  2071*  2073*  2081*
                        2083*  2088*  2091*  2118*  2123   2135*  2139*  2150*  2173   2201   2203*  2205*  2219*
                        2268*  2280*  2325*  2373*  2478*  2635   2638*  2655*  2703*  2790*  2801   2803   2808*
                        2812*  2813*  2896*  2897*  2957*  3018   3020*  3090*  3131*  3132*  3181*  3242   3244*
                        3326*  3327*  3383*  3443   3445*  3523*  3524*  3577*  3637   3639*  3789*  3796   3804*
                        3808*  3811*  3814*  3817*  3820*  3890*  3937*  3984*  4071*  4082   4084   4089*  4094*
                        4095*  4359*  4391   4393*  4454   4458*  4460   4648   4683   4692   4710   4717   4739
                        4741   5356*  5357
OR    = 000200           911#  1593
OUTBUF  030714          2934   2938   3092   3359   3363   3554   3558   3728   3758   3761   3840   4196   4199
                        5382#
PC    =%000007           825#  2129*  2133*  2144*  2148*  2184*  2192*  2199*  2814*  3022*  3246*  3506*  3702*
                        3706*  4101*  4397*  4401*  4519*  4537*  4554*  4556*  4560*  4562*  4564*  4566*  4568*
                        4570*  4572*  4574*  4576*  4578*  4580*  4582*  4584*  4586*  4588*  4590*  4672*  4733*
                        4754*  4767*  4778*  4816*  4830*  4843*  4856*  4864*  4970*  5153*  5254*  5266*  5380*
                        5408*  5414*  5420*  5435*
PCNT    001004           849#  4364*  4365*  4376
PGE   = 002000           912#
PGTRAP  005520          1657   1666#
PIP   = 020000           917#
PS    = 177776           815#   816   1118*  1183*  1659*  3935*
PSW   = 177776           816#
PTDONE  027720          5294   5330   5334   5352   5356#
Q0    = 000001          1114#  1178#  1229#  1250#  1259#  1288#  1304#  1316#  1325#  1341#  1355#  1362#  1378#
                        1392#  1399#  1412#  1418#  1424#  1431#  1441#  1448#  1456#  1464#  1480#  1492#  1504#
                        1516#  1543#  1570#  1616#  1652#  1671#  1849#  2024#  2111#  2151#  2206#  2256#  2363#
                        2416#  2468#  2518#  2565#  2640#  2885#  3120#  3310#  3509#  3709#  3777#  3827#  3879#
                        3921#  4177#  4282#
RBIT  = 104444          3009   3233   3434   3628   5173#
```

```
RDLIN = 104412          1006    5079    5160#
RDOCT = 104410          1012    5159#   5226
REGCHG  022136          4398    4403#
REPT    001204           943#   1710*   1719*   1739*   1748*   1752*   1761*   1766*   1775*   1809*   1811*   1840*   1842*
                        1881*   1893*   1918*   1925*   1941*   1981*   1992*   1996*   2003*   2234*   2239*   2279*   2289*
                        2300*   2302*   2309*   2314*   2324*   2334*   2702*   2712*   2728*   2737*   2754*   2763*   2834*
                        2843*   2956*   2966*   2981*   2990*   3006*   3016*   3030*   3053*   3062*   3093*   3101*   3180*
                        3190*   3206*   3215*   3230*   3240*   3254*   3277*   3286*   3382*   3392*   3407*   3416*   3431*
                        3441*   3451*   3473*   3483*   3576*   3586*   3601*   3610*   3625*   3635*   3645*   3667*   3677*
                        3983*   3993*   4009*   4018*   4035*   4044*   4120*   4129*   4200*   4202*   4210*   4218*   4235*
                        4244*   4304*   4308*   4319*   4323*   4685    4784*   4787*   4850*   5267*   5270*   5424*   5431*
REPT1   001206           944#   1808*   1838*   1942*   1949*   1991*   3005*   3021    3025*   3229*   3245    3249*   3430*
                        3624*   4688*   4690*   4694    4713*   4715*   4719    4797*   4800*   4808    4848*
RMRC1   011052          2372#
RMRC2   011236          2424#
RMRC3   011412          2477#
RMRC4   011564          2527#
RSAS    001116           871#   1034    1036    1051    1175    2167    2186    2249*   2250    2356*   2357    2540*   2616
                        3726*   5312
RSBA    001106           867#   1121*   1145    1187*   1204    1291*   1292    1295*   1296    1299*   1300    1309*   1310
                        1320*   1321*   1322    1506*   1508    1511*   1512    2686*   2934*   3164*   3359*   3554*   3728*
                        3758    3760    3840*   3967*   4196*   4303*   4318*   4343*   5317
RSBAB   001142           881#   1507*
RSCS1   001100           864#   1120*   1140    1185*   1199    1238*   1239    1242*   1243    1246*   1247    1253*   1254*
                        1255    1470*   1472    1475*   1476    1531    1534*   1535    1660*   1667    2053*   2125*   2141*
                        2175*   2178*   2229*   2246    2253    2304*   2353    2360    2379*   2386*   2400    2413    2430*
                        2451    2464    2484*   2505    2515    2533*   2552    2562    2598*   2611    2688*   2873    2936*
                        3076    3166*   3301    3361*   3498    3556*   3692    3730*   3731    3749    3801*   3815    3848*
                        3866    3903    3913    3969*   4164    4205*   4305*   4306    4311    4320*   4321    4326    4345*
                        4346    4353    5247    5297
RSCS1B  001134           878#   1471*   2619#
RSCS2   001102           865#   1026*   1045*   1049*   1060*   1119*   1126*   1131    1134*   1184*   1186*   1192*   1194*
                        1195    1198*   1265    1268*   1269    1271    1274*   1275    1278*   1279    1282*   1283*   1284
                        1287*   1482*   1483*   1487    1525    1538    1554    1581    1593    1605    2590*   2592    2600*
                        2604    2620    2624*   3740    3769    3895*   4302*   4317*   4342*   4348    5237*   5238*   5305
RSCS2B  001136           879#   1484*
RSDA    001110           868#   1122*   1151    1188*   1210    1366*   1367    1370*   1371    1374*   1375    1384*   1385
                        1394*   1395*   1396    2225*   2228*   2378*   2405    2429*   2437*   2456    2483*   2532*   2584*
                        2601*   2625    2879    3082    3307    3800*   3896*   3907    4167    4173    4194*   5322
RSDB    001122           873#   1190*   1524    1549*   1550*   1561    1566    1576*   1589    1601*   1611    1622*   1628*
                        1635    1643    5344
RSDS    001112           869#   1135    2054    2076    2131    2146    2194    3809    3862    3875    4259    4267    4440
                        5332
RSDT    001126           875#   1028    1061    1063    1065    1795    1821    1958    2226    5339
RSER    001114           870#   1027*   1046*   1123*   1157    1189*   1216    1403*   1404    1407*   1408    1414*   1415
                        1420*   1421    1426*   1427*   1428    2067    2084    2127    2142    2164*   2179    2390    2409
                        2441    2460    2491*   2495    2511    2543    2558    2591    3698    3725*   3802    3822    3855
                        3871    3898    3917    4100*   4170    4256    4264    5301
RSLA    001120           872#   1700    1732    1800    1832    1844    1890    1909    1920    1932    1944    1957    1998
                        2009    2016    5354
RSMR    001124           874#   1124*   1163    1193*   1221    1435*   1436    1443*   1444*   1445    1450*   1451    1458*
                        1459    2995*   2996*   3219*   3220*   3420*   3421*   3614*   3615*   4214*   4215*   4422*   4423*
                        4428    4437*   4438*   4520    4538    4594*   4595*   4629*   4636*   4677*   4697    4704*   4722
                        5243*   5272    5349
RSREG   027214          4970    5285#
RSVCPS  001132           877#   1658*   3966*
```

DZRSEC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
RSVEC   001130          876#  1657*  3965*
RSWC    001104          866#  1125*  1169   1191*  1226   1329*  1330   1333*  1334   1337*  1338   1347*  1348
                       1357*  1358*  1359   1494*  1496   1499*  1500   2687*  2876   2935*  3079   3165*  3304
                       3360*  3501   3555*  3695   3729*  3753   3841*  3968*  4195*  4300*  4316*  4341*  5327
RSWCB   001140          880#  1495*
RSO     001160          933#  4821*  4829   4842
R0      =%000000        818#  4369*  5039   5058*  5401*  5406*
R1      =%000001        819#  1044*  1047*  1574*  1575*  1576   1577   1599*  1600*  1601   1602   5040   5050*
                       5051*  5057*  5076   5080*  5093*  5095*  5097*  5100   5103*  5389*  5391*  5398*  5399*
R2      =%000002        820#  1017*  1019*  1024   1025   1738*  1797*  1799*  1800   1826*  1829*  1831*  1833
                       1867*  1910   1921   1931*  1933   1945   1954   1956*  1972*  1974*  1975*  1977   1986
                       1999   2042*  2053   2059   2094*  2095   2097   2099   2101   2103   2105   2107   2668*
                       2669*  2670*  2675*  2680*  2816*  2817*  2818*  2819   2912*  2913*  2914*  2919*  2923*
                       2937*  2940*  3024*  3026*  3027*  3028   3147*  3148*  3149*  3154*  3158*  3248*  3250*
                       3251*  3252   3346*  3348*  3351*  3352*  3354*  3362*  3366*  3446*  3447*  3448*  3449
                       3541*  3543*  3546*  3547*  3549*  3557*  3561*  3640*  3641*  3642*  3643   3790*  3798*
                       3800   3949*  3950*  3951*  3956*  3960*  4103*  4104*  4105*  4106   4212*  4214   5041
                       5056*  5077   5083*  5084   5086   5088   5090*  5091*  5092*  5094*  5096*  5102*  5259*
                       5260*  5263*  5388*  5390*  5394*  5395*  5397*  5402*  5403*  5405*  5422*  5426
R3      =%000003         821#  1986*  1987*  1988   2671*  2673*  2675   2677*  2681*  2915*  2917*  2919   2921*
                       2924*  2938*  2939*  3104*  3106   3109   3150*  3152*  3154   3156*  3159*  3347*  3349*
                       3353*  3355*  3363*  3365*  3542*  3544*  3548*  3550*  3558*  3560*  3952*  3954*  3956
                       3958*  3961*  4213*  4215   4508*  4515*  4518   4532*  4534*  4536   4673*  4676*  4677
                       4682   4702*  4703*  4704   4709   4860*  4861*  4863   5042   5055*  5078   5082*  5101*
                       5261*  5264*
R4      =%000004         822#  1013*  1014   1016   1020   1022*  2679*  2680   2784*  2922*  2923   3105*  3107
                       3110   3157*  3158   3959*  3960   4065*  4211*  4216*  4478*  4485*  4494*  4645*  4663*
                       4669*  4789*  4811*  4853*  4875   4885*  4889*  4903*  4994   4997*  5017   5023*  5043
                       5054*
R5      =%000005         823#  2783*  2784   2819*  2820*  3003*  3004*  3028*  3029*  3094*  3095*  3227*  3228*
                       3252*  3253*  3428*  3429*  3449*  3450*  3622*  3623*  3643*  3644*  4064*  4065   4106*
                       4107*  4199*  4201*  4477*  4478   4644*  4645   4788*  4789   4852*  4853   4876   4877*
                       4878   4880*  4881   4883   4891   4899*  4900*  4901   4902*  4995   4996*  5003*  5005*
                       5007*  5022*  5044   5053*  5116   5117*  5118   5138*  5423*  5429*
SAVEE   001174          940#  3021*  3025   3245*  3249   3335*  3448   3504*  3532*  3704*  4549
SC      = 100000        909#
SCOPE   = 104400       1117   1181   1232   1250   1262   1288   1304   1316   1325   1341   1355   1362   1378
                       1392   1399   1412   1418   1424   1431   1441   1448   1456   1467   1480   1492   1504
                       1519   1546   1570   1616   1655   1674   1852   2027   2114   2154   2209   2259   2366
                       2419   2471   2521   2568   2643   2888   3123   3313   3512   3712   3780   3830   3882
                       3924   4180   4285   4363   5155#
SEEC    027552         5308   5335#
SILO    005156         1573#
SILOB   004746         1523#
SILOFL  005330         1619#
SP      =%000006         824#   988*  3936*  4188*  4429   4430*  4433*  4441   4442*  4445*  4523*  4541*  4700*
                       4725*  5045   5049*  5187*  5188*  5193   5201   5202   5227   5253*  5275*  5386*
SPASS   = 104456       2693   2946   3171   3372   3567   3974   5178#
STTEST  002122         1055   1058#  1091
SUSWR   = 104460       5179#
SWI     026642         5185   5203*  5206#
SWR     001026          857#  1054   1069   1075   1096   3099   3112   4289   4366   4913   4915   4917   4928
                       4957   4961   4971   4974   5190   5194*  5196   5212
SWREG   000176          799#  5194   5196   5212   5223   5230*
SW10    = 002000        787#  4366   4956   4957
SW11    = 004000        786#  4928
```

```
SW12  = 010000            785#   4367   4369
SW13  = 020000            784#   4961
SW14  = 040000            783#   4917
SW15  = 100000            782#
SW8   = 000400            790#   4912   4913
SW9   = 001000            789#   4974
TABTYP  031736           5414    5420   5422#
TBDIA   021362           4314    4332#
TIMES   025404           1112    1113*  1237*  4932   4947#
TIMSV   001172            938#   1112*  1237
TKB     001022            855#   5216
TKS     001020            854#   5214
TPB     001024            856#   4886*  4891*
TPS     001016            853#   4887   4892
TRE   = 040000            908#   1534
TRMR    022173           2394    2445   2499   4409#
TRY     002016           1046#   1050
TRYNX   002352           1059    1066   1086#  4360
TSTEVB  022704           4510    4513   4517#
TST1    002606           1117#
TST10   003646           1316#
TST11   003672           1325#
TST12   003754           1341#
TST13   004006           1355#
TST14   004032           1362#
TST15   004114           1378#
TST16   004146           1392#
TST17   004172           1399#
TST2    003030           1181#
TST20   004234           1412#
TST21   004256           1418#
TST22   004300           1424#
TST23   004324           1431#
TST24   004362           1441#
TST25   004410           1448#
TST26   004446           1456#
TST27   004504           1467#
TST3    003252           1232#
TST30   004556           1480#
TST31   004622           1492#
TST32   004672           1504#
TST33   004744           1519#
TST34   005046           1546#
TST35   005154           1570#
TST36   005326           1616#
TST37   005442           1655#
TST4    003344           1250#
TST40   005534           1674#
TST41   006270           1852#
TST42   007044           2027#
TST43   007566           2114#
TST44   007756           2154#
TST45   010330           2209#
TST46   010512           2259#
TST47   011050           2366#
TST5    003372           1262#
```

```
TST50   011234          2419#
TST51   011410          2471#
TST52   011562          2521#
TST53   011776          2568#
TST54   012210          2643#
TST55   013242          2888#
TST56   014230          3123#
TST57   015034          3313#
TST6    003532          1288#
TST60   015662          3512#
TST61   016560          3712#
TST62   017000          3780#
TST63   017176          3830#
TST64   017512          3882#
TST65   017632          3924#
TST66   020632          4180#
TST67   021136          4285#
TST7    003614          1304#
TTAGG   003032          1183#
TYPE  = 104402          1005    1011    1030    1032    1056    1071    1074    1079    1083    1100    1101    2058    2080
                        2089    2170    2183    2191    2198    2394    2395    2445    2446    2499    2500    2547    2637
                        3114    3701    3859    3873    4333    4350    4368    4381    4394    4398    4555    4559    4561
                        4563    4565    4567    4569    4571    4573    4575    4577    4579    4581    4583    4585    4587
                        4589    4959    4963    4969    5021    5061    5107    5126    5131    5134    5137    5156#   5220
                        5222    5225    5287    5290    5296    5300    5304    5311    5316    5321    5326    5331    5338
                        5343    5348    5353    5359    5379    5411    5417    5428    5433
TYPE0 = 104404          4968    5157#   5224    5289    5292    5298    5302    5306    5313    5318    5323    5328    5333
                        5340    5345    5350    5355    5413    5419    5427
TYPES = 104406          1073    1082    1099    2061    3116    4396    4400    5158#   5366    5376
UNCMP   001166           936#   1025*   1052*   1058    1088*   1094*   1102*   2165    2187
UNITSV  001164           935#   1024*   1034    1037    1051*   1058    2249    2356    3[illegible]
UNNUM   001162           934#   1016*   1026    1053*   1060    1072    1081    1090*   1095*   1098    1119    1134    1184
                        1198    1287    1482    1485    1527    1553    1580    2624    2631    3742    3771    5238
WAITRY  027054          5246#
WC    = 000010           896#   1171    1228    1332    1336    1340    1361    1498    1502    2875    2878    3078    3081
                        3303    3306    3500    3503    3694    3697    3700    3755    3868    4158    4166    4172    4313
                        4328    5324
WCE   = 040000           914#   4348
WCRC    001202           942#   4801*   4826    4839
WK15    001214           947#   4802*   4805*   4815
WORK    001216           948#   1436*   1437*   1438    1451*   1452*   1453    1459*   1460*   1461    1551*   1557*   1661*
                       1662*   2059*   2060    2126*   2140*   2172*   2177*   2588*   2595*   2785*   2805*   2809*   2810*
                        2818    2821*   2822*   3027    3111*   3115    3251    3735*   3736*   4066*   4086*   4090*   4091*
                        4105    4108*   4109*   4338*   4339*   4395    4399    4547*   4551*   4554    4676    4702    4729*
                        4732*   4815*   4825*   4828*   4838*   4841*   4855*   4860    4862    5246*   5249*   5250    5409
                        5412    5415    5418
WORK0   001220           949#   4790*   4801    4804*   4817*   4820*   4831*   4834*   4844*   4847*   4855
WORK1   001222           950#   2599*   2630    4548*   4549    4552*
WORK2   001224           951#   5361*   5362*   5369*   5371*   5373*
WORK3   001226           952#   3002*   3226*   3427*   3621*   4468*   4473    4487*   4642    4652*   4656*   4665*   5387*
                        5393*   5396*   5397
WORK4   001230           953#   5360*   5363*   5365
WORK5   001232           954#   4294*   4295*   5367*   5377*
WORK6   001234           955#   5368*   5370*   5372*   5374*   5375
XBIT  = 104432          2791    4072    5168#
XXOR    025030          4816    4830    4843    4860#
```

DZRSEC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
XXX     026632          5186    5203#
ZERONE  005050          1548#
$ENDAD  021736           793    4372#
$END1   021746          4370    4374#
.     = 032012           791#    792#    794#    797#    801#    804#    845#    860#   1005   1006#   1011   1030   1031#
                        1032    1033#   1056    1071    1072#   1074    1079    1101   1102#   1132   1136   1141   1146
                        1152    1158    1164    1170    1176    1196    1200    1205   1211    1217   1222   1227   1240
                        1244    1248    1256    1266    1276    1280    1285    1293   1297    1301   1312   1323   1331
                        1335    1339    1350    1360    1368    1372    1376    1387   1397    1405   1409   1416   1422
                        1429    1439    1446    1454    1462    1473    1477    1489   1497    1501   1509   1513   1529
                        1532    1536    1541    1563    1568    1583    1594    1603   1606    1614   1624   1637   1640
                        1645    1668    1701    1847    1891    2058    2080    2081#  2089    2090#  2170   2171#  2183
                        2191    2198    2254    2361    2395    2414    2446    2447#  2465    2500   2501#  2516   2547
                        2548#   2563    2608    2612    2617    2621    2637    2877   2880    2883   3080   3083   3114
                        3305    3308    3502    3696    3701    3702#   3754    3773   3823    3859   3873   3877   3899
                        3904    3908    3914    3918    4165    4168    4171    4174   4268    4333   4350   4351#  4354
                        4368    4369#   4376    4381    4394    4416#   4555    4888   4893    4947   4963   4964#  4969
                        4970#   4972    4980    4981    4990    5010    5013    5018   5028#   5061   5065   5107   5121
                        5126    5127#   5131    5141#   5220    5221#   5222    5225   5226#   5287   5290   5291#  5296
                        5300    5301#   5304    5311    5312#   5316    5317#   5321   5322#   5326   5327#  5331   5332#
                        5338    5339#   5343    5344#   5348    5349#   5353    5354#  5359    5360#  5379   5380#  5381#
                        5382#   5411    5417    5433    5434#
.CLKDO  022734          4529#   5170
.CLKD1  022660          4507#   5169
.CLKRO  024046          4702#   5172
.CLKR1  023664          4672#   5171
.CLRDK  027016          5161    5237#
.CNTLU  026726          5180    5182    5221#
.DSCK   022342          4440#   5166
.GETSP  027136          5177    5267#
.HLT    025406           991    4956#
.HLTCT  025544          4966*   4980#   5285    5307    5309    5314    5319    5324   5329    5335*  5336   5341   5346
                        5351
.ILLUP  026106          5037    5064#
.KBDIN  026644          5181    5210#
.KIT    025356          4918    4927    4933    4940#   4978
.MCLKB  023256          4594#   5176
.MCLKO  023500          4636#   5175
.MCLK1  023456          4629#   5174
.MRCK   022276          4428#   5163
.MRCLK  022246          4422#   5164
.MRDMD  027044          5162    5243#
.MRIND  022324          4437#   5167
.MRINT  022240          4418#   5165
.MSG    027012          5216*   5217*   5218    5227*   5230    5233#
.OVER   025362          4916    4941#
.POWER  025760           989    5037#   5059
.POWUP  026024          5046    5049#
.PR     025734          4987*   4989*   4991*   4992*   4997    5001    5011*   5012   5015*   5017   5021   5028#
.PRF    025642          4999    5002    5007#
.PRL    025620          5000#   5016
.PTIT   025600          4988    4993#
.PUVEC  026114          5037*   5038*   5046*   5068#
.QQ.  = 000070          5142#   5155    5156#   5157#   5158#   5159#   5160#   5161#  5162#   5163#  5164#  5165#  5166#
                        5167#   5168#   5169#   5170#   5171#   5172#   5173#   5174#  5175#   5176#  5177#  5178#  5179#
                        5180#   5181#   5182#   5183#
```

DZRSEC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
.RBIT   023510          4642#   5173
.RDLIN  026252          5116#   5160
.RDOCT  026120          5074#   5159
.SAVR6  026112          5045*   5049    5067#
.SCOPE  025212          4912#   5155
.SPASS  027176          5178    5278#
.SUSWR  026524          5179    5185#
.SVLAD  025336          4924    4929    4935#   4943
.SWOPT= 163400           775#    782     783     784     785     786     787     788     789     790     791    4366    4367
                        4369    4912    4956    4961    4974
.TBIT   021752          4376#
.TRAP   026414           993    5149#
.TRP    026432          5152    5153#
.TYPE   025054          4875#   5156
.TYPEB  025546          4987#
.TYPEO  025556          4989#   5157
.TYPES  025566          4991#   5158
.XBIT   022370          4450#   5168
```

DZRSEC.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
END          1#    775#
NEWTST       1#    775#  1114  1178  1229  1250  1259  1288  1304  1316  1325  1341  1355  1362  1378
          1392  1399  1412  1418  1424  1431  1441  1448  1456  1464  1480  1492  1504  1516  1543
          1570  1616  1652  1671  1849  2024  2111  2151  2206  2256  2363  2416  2468  2518  2565
          2640  2885  3120  3310  3509  3709  3777  3827  3879  3921  4177  4282
ODT11        1#
ODT11X       1#
POP          1#    842#  4923  4926  5022  5053  5101
PRINT        1#    842#  1005  1011  1030  1032  1056  1071  1074  1079  1101  2058  2080  2089  2170
          2183  2191  2198  2395  2446  2500  2547  2637  3114  3701  3859  3873  4333  4350  4368
          4381  4394  4555  4963  4969  5061  5106  5126  5130  5220  5222  5225  5287  5290  5296
          5300  5304  5311  5316  5321  5326  5331  5338  5343  5348  5353  5359  5379  5411  5417
          5433
PUSH         1#    842#  4920  4993  5039  5076
SCOPE.       1#   1114#  1178#  1229#  1250#  1259#  1288#  1304#  1316#  1325#  1341#  1355#  1362#  1378#  1392#
          1399#  1412#  1418#  1424#  1431#  1441#  1448#  1456#  1464#  1480#  1492#  1504#  1516#  1543#  1570#
          1616#  1652#  1671#  1849#  2024#  2111#  2151#  2206#  2256#  2363#  2416#  2468#  2518#  2565#  2640#
          2885#  3120#  3310#  3509#  3709#  3777#  3827#  3879#  3921#  4177#  4282#
SET       5142#  5155  5156  5157  5158  5159  5160  5161  5162  5163  5164  5165  5166  5167  5168
          5169  5170  5171  5172  5173  5174  5175  5176  5177  5178  5179  5180  5181  5182
TYPADR       1#
TYPBIT       1#    842#
TYPCHR       1#    842#
TYPDEC       1#    842#
TYPLIN       1#    842#
TYPOCS       1#    842#  1072  1080  1098  2060  3115  4395  4399  5365  5375
TYPOCT       1#    842#  4967  5223  5288  5291  5297  5301  5305  5312  5317  5322  5327  5332  5339
          5344  5349  5354  5412  5418  5425
TYPTXT       1#    842#  1004  1010  1030  1031  1056  1071  1074  1079  1101  2058  2080  2089  2170
          2183  2191  2198  2395  2446  2500  2547  2637  3114  3701  3859  3873  4332  4350  4380
          4394  4555  5220  5221  5225  5287  5290  5295  5299  5303  5311  5316  5321  5326  5331
          5338  5343  5348  5353  5359  5379  5411  5417  5433
$CATCH       1#    775#   791
$CMTAG       1#    775#   845
$DONE        1#    775#  4361
$EQUAT       1#    775#   813
$HLT         1#    775#  4948
$KBDIN       1#    775#  5209
$KMMR        1#
$OCTAL       1#    775#  4981
$POWER       1#    775#  5029
$RAND2       1#
$RAND4       1#
$RDDEC       1#
$RDLIN       1#    775#  5109
$RDOCT       1#    775#  5069
$SCOPE       1#    775#  4905
$SET      5142#  5155  5156  5157  5158  5159  5160  5161  5162  5163  5164  5165  5166  5167  5168
          5169  5170  5171  5172  5173  5174  5175  5176  5177  5178  5179  5180  5181  5182
$SETUP       1#    775#   988
$SMMR        1#
$SUSWR       1#    775#  5184
$SWDOC       1#    775#   779
$TRAP        1#    775#  5142
$TYPE        1#    775#  4865
$TYPEA       1#
```

```
STYPED      1#
SUMMR       1#
```

```
ADC     4365  4425  4597  4631  4675
ADD     1797  1799  1829  1831  1974  2094  2817  3026  3250  3447  3641  4104  4364  4424  4430
        4433  4442  4445  4477  4523  4541  4551  4596  4630  4644  4674  4700  4725  4852  4896
        4899  5152  5253  5260  5275  5395  5396  5403
ASL     1008
BCC     1315  1353  1390
BCS     1089  1624  1630  1640  1649  2596  2674  2918  3153  3505  3705  3799  3955
BEQ     1002  1008  1021  1035  1048  1059  1062  1064  1078  1132  1136  1141  1146  1152  1158
        1164  1170  1176  1196  1200  1205  1211  1217  1222  1227  1240  1244  1248  1256  1266
        1270  1276  1280  1285  1293  1297  1301  1312  1323  1331  1335  1339  1350  1360  1368
        1372  1376  1387  1397  1405  1409  1416  1422  1429  1439  1446  1454  1462  1473  1477
        1489  1497  1501  1509  1513  1529  1532  1536  1541  1556  1563  1568  1583  1588  1591
        1594  1603  1614  1637  1645  1668  1701  1733  1796  1801  1822  1824  1835  1847  1891
        1912  1923  1935  1947  1955  1959  1961  1979  1989  2001  2012  2019  2057  2070  2079
        2087  2096  2098  2100  2102  2104  2106  2108  2128  2132  2143  2147  2169  2182  2190
        2197  2227  2247  2251  2254  2354  2358  2361  2393  2401  2406  2410  2414  2444  2452
        2457  2461  2465  2498  2506  2512  2516  2546  2553  2559  2563  2608  2612  2617  2621
        2628  2877  2880  3077  3080  3083  3096  3098  3102  3302  3305  3308  3499  3502  3693
        3696  3699  3732  3744  3750  3754  3759  3773  3803  3810  3816  3823  3858  3863  3867
        3872  3877  3899  3904  3908  3914  3918  4165  4168  4171  4174  4257  4260  4265  4268
        4349  4370  4451  4531  4651  4659  4680  4684  4686  4693  4707  4711  4718  4731  4750
        4786  4799  4819  4827  4833  4840  4846  4851  4882  4914  4916  4931  4943  4958  5010
        5013  5085  5119  5191  5229  5251  5273  5310  5315  5320  5325  5330  5337  5342  5347
        5352  5358  5400  5430
BGT     4933  5087
BHIS    4758
BIC      808   997   999  1000  1057  1084  1437  1452  1460  1539  1825  1865  1971  1972  1976
        1987  2039  2066  2073  2083  2091  2118  2135  2150  2166  2188  2205  2373  2478  2813
        2897  3090  3132  3327  3524  3789  3808  3814  3820  3890  4095  4456  4458  4466  4467
        4499  4534  4653  4660  4687  4703  4712  4753  4761  4766  4771  4774  4777  4817  4831
        4844  4861  4862  4900  5217  5335
BICB    5123
BIS      805  1009  1111  1483  1486  1527  1553  1580  1686  1828  1864  1969  1993  2062  2071
        2081  2088  2139  2203  2219  2268  2280  2325  2605  2638  2703  2790  2808  2812  2896
        2957  3020  3131  3181  3244  3326  3364  3383  3445  3523  3559  3577  3639  3742  3771
        3804  3811  3817  3895  3984  4071  4089  4094  4302  4317  4342  4359  4393  4479  4498
        4515  4518  4532  4536  4646  4676  4681  4682  4691  4696  4702  4708  4709  4716  4721
        4732  4748  4820  4828  4834  4841  4847  4863  5203  5356
BISB    5014
BIT     1001  1054  1058  1069  1075  1077  1086  1092  1096  1163  1199  1216  1221  1445  1535
        1593  1605  1823  1960  1967  2040  2123  2173  2201  2592  2620  2635  2801  2803  2882
        3018  3099  3112  3242  3333  3443  3530  3637  3731  3796  4082  4084  4289  4348  4366
        4384  4391  4450  4454  4460  4464  4471  4475  4482  4496  4549  4648  4650  4658  4679
        4683  4692  4706  4710  4717  4739  4741  4785  4798  4806  4826  4839  4878  4913  4917
        4928  4957  4961  4974  5001  5185  5272  5307  5309  5314  5319  5324  5329  5336  5341
        5346  5351  5357
BLOS    1015  4744  4747  4809
BLT     5089
BMI     4307  4322  5248
BNE     1029  1055  1066  1068  1070  1076  1087  1093  1097  1558  1578  1606  1663  1720  1749
        1762  1776  1812  1839  1843  1894  1926  1950  1968  1982  2004  2041  2124  2174  2202
        2240  2290  2303  2315  2335  2593  2636  2682  2713  2738  2764  2802  2804  2806  2811
        2823  2844  2883  2925  2941  2967  29°!  3017  3019  3063  3100  3113  3160  3191  3216
        3241  3243  3287  3334  3350  3356  336,  3393  3417  3442  3444  3484  3531  3545  3551
        3562  3587  3611  3636  3638  3678  3739  3797  3962  3994  4019  4045  4083  4085  4087
        4092  4110  4130  4203  4217  4219  4245  4290  4296  4309  4324  4340  4367  4385  4392
```

```
        4432  4444  4455  4461  4465  4472  4474  4476  4483  4497  4510  4513  4522  4540  4550
        4643  4649  4689  4695  4699  4714  4720  4724  4740  4742  4752  4760  4764  4773  4776
        4807  4849  4879  4884  4890  4898  4918  4929  4962  4975  5002  5016  5018  5052  5125
        5129  5136  5186  5197  5199  5211  5213  5219  5265  5271  5286  5308  5378  5392  5407
        5410  5416  5432
BPL     2874  4312  4327  4347  4354  4888  4893  4972  5121  5215
BR       809  1023  1050  1085  1091  1104  1604  1625  1641  1798  1827  1830  1970  1973  2176
        2597  2633  2676  2920  3031  3118  3155  3255  3452  3646  3825  3957  4159  4314  4352
        4480  4533  4553  4654  4661  4671  4745  4762  4765  4813  4854  4894  4919  4924  4927
        4988  4990  4999  5065  5099  5108  5127  5132  5192  5252  5293  5421  5434
CLC     1308  1346  1383  1620  1633  3703  4484  4803  5262  5364  5404
CLR      995   996   998  1017  1045  1095  1254  1283  1321  1358  1395  1427  1444  1549  1560
        1574  1585  1599  1738  1866  1867  2017  2085  2437  2540  2589  2601  2669  2671  2904
        2913  2915  2939  3002  3148  3150  3226  3348  3354  3365  3427  3543  3549  3560  3621
        3950  3952  4229  4382  4383  4459  4462  4481  4493  4508  4647  4655  4662  4668  4729
        4790  4802  4810  4814  4825  4838  4991  5050  5080  5081  5239  5240  5246  5280  5281
        5368  5387  5390
CLRB    1484  4976  4998  5000  5020
CMP     1014  1028  1034  1061  1063  1065  1131  1135  1140  1169  1195  1226  1239  1243  1247
        1255  1265  1269  1275  1279  1284  1292  1296  1300  1311  1330  1334  1338  1349  1367
        1371  1375  1386  1404  1408  1415  1421  1438  1453  1461  1472  1476  1488  1496  1500
        1508  1512  1528  1531  1540  1555  1562  1567  1577  1582  1587  1590  1602  1613  1636
        1644  1666  1667  1795  1800  1821  1834  1846  1890  1911  1922  1934  1946  1954  1958
        1978  1988  2000  2011  2018  2056  2069  2078  2086  2095  2097  2099  2101  2103  2105
        2107  2131  2146  2168  2181  2189  2196  2226  2246  2253  2353  2360  2392  2400  2405
        2413  2443  2451  2456  2464  2497  2505  2515  2545  2552  2562  2607  2611  2625  2879
        3076  3082  3097  3103  3117  3301  3307  3498  3692  3698  3743  3749  3753  3758  3772
        3802  3809  3815  3857  3862  3866  3875  3898  3903  3907  3913  4164  4167  4170  4173
        4256  4259  4267  4431  4443  4521  4539  4685  4694  4698  4719  4723  4743  4746  4749
        4751  4757  4759  4763  4772  4775  4808  4897  4925  5017  5118  5190  5193  5196  5212
        5409  5415
CMPB    1007  4883  4915  4932  5084  5086  5088  5124  5128  5135  5218
DEC     1022  1047  1557  1662  1719  1748  1761  1775  1811  1838  1842  1893  1925  1949  1981
        2003  2239  2289  2302  2314  2334  2681  2712  2737  2763  2805  2810  2822  2843  2924
        2940  2966  2990  3016  3062  3095  3101  3159  3190  3215  3240  3286  3349  3355  3366
        3392  3416  3441  3483  3544  3550  3561  3586  3610  3635  3677  3736  3961  3993  4018
        4044  4086  4091  4109  4129  4202  4216  4218  4244  4295  4308  4323  4339  4487  4665
        4688  4713  4848  4850  4889  5264  5270  5377  5391  5399  5406  5429  5431
EMT      814
HALT     792  1033  1103  4973  5047  5064
INC     1049  1090  1575  1586  1600  1931  1956  2594  4738  4960  5051  5098  5249
INCB    4935  4940  5011  5015
JMP      802   806  1003  1041  1106  1665  1990  1994  2109  2204  2884  3507  3707  4291  4334
        4360  4374  4386  4387  4978  5062  5294  5334
JSR     2129  2133  2144  2148  2184  2192  2199  2814  3022  3246  3506  3702  3706  4101  4372
        4519  4537  4554  4672  4816  4830  4843  4970  5408  5414  5420
MOV      988   989   990   991   992   993   994  1013  1016  1024  1025  1026  1027  1036  1037
        1044  1046  1051  1052  1053  1060  1072  1081  1094  1098  1102  1105  1112  1113  1118
        1119  1120  1121  1122  1123  1124  1125  1126  1134  1183  1184  1185  1186  1187  1188
        1189  1190  1191  1192  1193  1194  1198  1237  1238  1242  1246  1253  1268  1271  1272
        1274  1278  1282  1287  1291  1295  1299  1307  1309  1310  1320  1329  1333  1337  1345
        1347  1348  1357  1366  1370  1374  1382  1384  1385  1394  1403  1414  1420  1426  1435
        1436  1443  1450  1451  1458  1459  1470  1482  1485  1487  1494  1506  1524  1525  1526
        1534  1538  1550  1551  1552  1554  1561  1565  1566  1576  1579  1581  1589  1601  1611
        1612  1621  1622  1626  1628  1634  1635  1642  1643  1657  1658  1659  1660  1661  1710
        1739  1752  1766  1808  1809  1832  1833  1840  1844  1845  1881  1909  1910  1918  1920
```

```
            1921  1932  1933  1941  1942  1944  1945  1957  1977  1986  1991  1992  1996  1998  1999
            2009  2010  2016  2042  2053  2054  2055  2059  2060  2067  2068  2076  2077  2084  2125
            2126  2140  2141  2164  2165  2167  2172  2175  2177  2178  2179  2180  2186  2187  2194
            2195  2225  2228  2229  2234  2249  2279  2300  2304  2309  2324  2356  2378  2379  2386
            2390  2391  2429  2430  2441  2442  2483  2484  2491  2495  2496  2532  2533  2543  2544
            2584  2588  2590  2598  2599  2600  2604  2624  2630  2631  2653  2655  2668  2670  2675
            2677  2679  2680  2686  2687  2688  2702  2728  2754  2783  2784  2785  2809  2816  2818
            2819  2821  2834  2912  2914  2919  2921  2922  2923  2934  2935  2936  2937  2938  2956
            2981  2995  2996  3003  3005  3006  3021  3024  3025  3027  3028  3030  3053  3091  3092
            3093  3094  3104  3105  3106  3107  3109  3110  3111  3115  3139  3147  3149  3154  3156
            3157  3158  3164  3165  3166  3180  3206  3219  3220  3227  3229  3230  3245  3248  3249
            3251  3252  3254  3277  3324  3335  3346  3347  3351  3352  3353  3359  3360  3361  3362
            3363  3382  3407  3420  3421  3428  3430  3431  3446  3448  3449  3451  3473  3521  3532
            3541  3542  3546  3547  3548  3554  3555  3556  3557  3558  3576  3601  3614  3615  3622
            3624  3625  3640  3642  3643  3645  3667  3725  3726  3728  3729  3730  3735  3740  3741
            3760  3761  3769  3770  3790  3791  3800  3801  3840  3841  3848  3855  3856  3896  3933
            3935  3936  3937  3949  3951  3956  3958  3959  3960  3965  3966  3967  3968  3969  3983
            4009  4035  4064  4065  4066  4090  4100  4103  4105  4106  4108  4120  4188  4194  4195
            4196  4199  4200  4201  4205  4210  4211  4212  4213  4214  4215  4235  4292  4294  4300
            4301  4303  4304  4305  4316  4318  4319  4320  4338  4341  4343  4344  4345  4369  4395
            4399  4422  4423  4428  4429  4437  4438  4440  4441  4452  4457  4468  4470  4478  4517
            4520  4535  4538  4547  4548  4594  4595  4629  4636  4645  4652  4656  4673  4677  4678
            4690  4697  4704  4705  4715  4722  4784  4787  4788  4789  4797  4800  4801  4815  4821
            4829  4842  4853  4855  4860  4875  4876  4877  4880  4895  4901  4902  4903  4920  4921
            4923  4926  4936  4937  4941  4944  4964  4967  4987  4994  4995  4996  4997  5022  5023
            5024  5025  5037  5038  5039  5040  5041  5042  5043  5044  5045  5046  5049  5053  5054
            5055  5056  5057  5058  5059  5060  5074  5075  5076  5077  5078  5082  5100  5101  5102
            5103  5116  5117  5138  5149  5151  5153  5187  5188  5189  5194  5195  5201  5202  5216
            5223  5227  5230  5237  5238  5243  5259  5261  5267  5288  5291  5297  5301  5305  5312
            5317  5322  5327  5332  5339  5344  5349  5354  5360  5361  5365  5367  5375  5385  5386
            5388  5389  5393  5394  5397  5398  5401  5402  5412  5418  5422  5423  5424  5426
MOVB        1407  1471  1475  1495  1499  1507  1511  2619  4885  4886  4891  4934  4966  4989  4392
            5019  5083  5122  5133
NOP         3737  3738  4157  4176  4356
RESET       1127  1264  4371
ROL         1019  1314  1352  1389  1623  1629  1639  1648  2595  2673  2917  3152  3504  3704  3798
            3954  4485  4486  4494  4495  4552  4663  4664  4669  4670  4804  4805  4811  4812  5003
            5005  5007  5093  5094  5095  5096  5097  5263  5362  5363  5369  5370  5371  5372  5373
            5374  5405
ROLB        5004  5006  5008
ROR         5090  5091  5092
RTI         4420  4426  4434  4439  4446  4469  4488  4500  4524  4542  4598  4632  4637  4657  4666
            4701  4726  4904  4938  4945  4977  5026  5104  5139  5204  5231  5241  5244  5274  5276
            5282
RTS         4397  4401  4556  4560  4562  4564  4566  4568  4570  4572  4574  4576  4578  4580  4582
            4584  4586  4588  4590  4733  4754  4767  4778  4856  4864  5254  5266  5380  5435
SEC         1018  1627  1647  2672  2916  3151  3953
SUB         1826  1975  2820  3004  3029  3228  3253  3429  3450  3623  3644  4107  4965  5150
TRAP        5152  5155  5156  5157  5158  5159  5160  5161  5162  5163  5164  5165  5166  5167  5168
            5169  5170  5171  5172  5173  5174  5175  5176  5177  5178  5179  5180  5181  5182
TST         1020  1067  1145  1151  1157  1175  1204  1210  1322  1359  1396  1428  1700  1732  2127
            2142  2250  2357  2409  2460  2511  2558  2591  2616  2873  2876  3079  3304  3501  3695
            3822  3871  3917  4264  4311  4326  4353  4473  4509  4512  4530  4642  4730  4818  4832
            4845  4922  4942  4971  5198  5210  5228  5250  5285
TSTB        4306  4321  4346  4881  4887  4892  4930  5009  5012  5120  5214  5247
.ASCIZ      1006  1012  1031  1033  1057  1072  1075  1080  1102  2059  2081  2090  2171  2184  2192
```

```
            2199    2396    2447    2501    2548    2638    3115    3702    3860    3874    4334    4351    4369    4382    4395
            4403    4409    4556    4599    4600    4601    4602    4604    4606    4608    4610    4612    4614    4616    4618
            4620    4621    4622    4624    4964    4970    5062    5108    5127    5132    5221    5223    5226    5288    5291
            5297    5301    5305    5312    5317    5322    5327    5332    5339    5344    5349    5354    5360    5380    5412
            5418    5434
.BLKB       5141
.BLKW       5028    5381    5382
.ENABL         1     775
.END        5437
.ENDC        842     989     995    1117    1181    1232    1250    1262    1288    1304    1316    1325    1341    1355    1362
            1378    1392    1399    1412    1418    1424    1431    1441    1448    1456    1467    1480    1492    1504    1519
            1546    1570    1616    1655    1674    1852    2027    2114    2154    2209    2259    2366    2419    2471    2521
            2568    2643    2888    3123    3313    3512    3712    3780    3830    3882    3924    4180    4285    4366    4368
            4369    4375    4918    4946    4961    4963    4971    4979    5045    5053    5062    5063    5119    5142    5156
            5157    5158    5159    5160    5161    5162    5163    5164    5165    5166    5167    5168    5169    5170    5171
            5172    5173    5174    5175    5176    5177    5178    5179    5180    5181    5182    5183
.EVEN       1006    1012    1031    1033    1057    1072    1075    1080    1102    2059    2081    2090    2171    2184    2192
            2199    2396    2447    2501    2548    2638    3115    3702    3860    3874    4334    4351    4369    4382    4395
            4416    4556    4964    4970    5062    5108    5127    5132    5221    5223    5226    5288    5291    5297    5301
            5305    5312    5317    5322    5327    5332    5339    5344    5349    5354    5360    5380    5412    5418    5434
.IF          826     988     995    1114    1178    1229    1250    1259    1288    1304    1316    1325    1341    1355    1362
            1378    1392    1399    1412    1418    1424    1431    1441    1448    1456    1464    1480    1492    1504    1516
            1543    1570    1616    1652    1671    1849    2024    2111    2151    2206    2256    2363    2416    2468    2518
            2565    2640    2885    3120    3310    3509    3709    3777    3827    3879    3921    4177    4282    4366    4367
            4369    4374    4912    4946    4956    4961    4970    4974    5045    5053    5061    5062    5118    5141    5155
            5156    5157    5158    5159    5160    5161    5162    5163    5164    5165    5166    5167    5168    5169    5170
            5171    5172    5173    5174    5175    5176    5177    5178    5179    5180    5181    5182
.IFF         989    4367    4374    4918    4956    4974    5062    5119    5142    5155    5156    5157    5158    5159    5160
            5161    5162    5163    5164    5165    5166    5167    5168    5169    5170    5171    5172    5173    5174    5175
            5176    5177    5178    5179    5180    5181    5182
.IIF         782     783     784     785     786     787     788     789     790     791    4376    4938    4942    4947    4980
            4981    5141
.IRP        4920    4923    4926    4994    5022    5039    5053    5076    5101    5155
.LIST          1     775     792     842    1006    1012    1031    1033    1057    1072    1075    1080    1102    1114    1118
            1178    1182    1229    1233    1250    1251    1259    1263    1288    1289    1304    1305    1316    1317    1325
            1326    1341    1342    1355    1356    1362    1363    1378    1379    1392    1393    1399    1400    1412    1413
            1418    1419    1424    1425    1431    1432    1441    1442    1448    1449    1456    1457    1464    1468    1480
            1481    1492    1493    1504    1505    1516    1520    1543    1547    1570    1571    1616    1617    1652    1656
            1671    1675    1849    1853    2024    2028    2059    2081    2090    2111    2115    2151    2155    2171    2184
            2192    2199    2206    2210    2256    2260    2363    2367    2396    2416    2420    2447    2468    2472    2501
            2518    2522    2548    2565    2569    2638    2640    2644    2885    2889    3115    3120    3124    3310    3314
            3509    3513    3702    3709    3713    3777    3781    3827    3831    3860    3874    3879    3883    3921    3925
            4177    4181    4282    4286    4334    4351    4369    4382    4395    4556    4964    4970    5062    5108    5127
            5132    5142    5155    5156    5157    5158    5159    5160    5161    5162    5163    5164    5165    5166    5167
            5168    5169    5170    5171    5172    5173    5174    5175    5176    5177    5178    5179    5180    5181    5182
            5183    5221    5223    5226    5288    5291    5297    5301    5305    5312    5317    5322    5327    5332    5339
            5344    5349    5354    5360    5380    5412    5418    5434
.MACRO         1    5142
.MCALL       775     842    1114    1178    1229    1250    1259    1288    1304    1316    1325    1341    1355    1362    1378
            1392    1399    1412    1418    1424    1431    1441    1448    1456    1464    1480    1492    1504    1516    1543
            1570    1616    1652    1671    1849    2024    2111    2151    2206    2256    2363    2416    2468    2518    2565
            2640    2885    3120    3310    3509    3709    3777    3827    3879    3921    4177    4282
.NLIST         1     775     792     842    1006    1012    1031    1033    1057    1072    1075    1080    1102    1114    1118
            1178    1182    1229    1233    1250    1251    1259    1263    1288    1289    1304    1305    1316    1317    1325
            1326    1341    1342    1355    1356    1362    1363    1378    1379    1392    1393    1399    1400    1412    1413
            1418    1419    1424    1425    1431    1432    1441    1442    1448    1449    1456    1457    1464    1468    1480
```

```
            1481    1492    1493    1504    1505    1516    1520    1543    1547    1570    1571    1616    1617    1652    1656
            1671    1675    1849    1853    2024    2028    2059    2081    2090    2111    2115    2151    2155    2171    2184
            2192    2199    2206    2210    2256    2260    2363    2367    2396    2416    2420    2447    2468    2472    2501
            2518    2522    2548    2565    2569    2638    2640    2644    2885    2889    3115    3120    3124    3310    3314
            3509    3513    3702    3709    3713    3777    3781    3827    3831    3860    3874    3879    3883    3921    3925
            4177    4181    4282    4286    4334    4351    4369    4382    4395    4556    4964    4570    5062    5108    5127
            5132    5142    5155    5156    5157    5158    5159    5160    5161    5162    5163    5164    5165    5166    5167
            5168    5169    5170    5171    5172    5173    5174    5175    5176    5177    5178    5179    5180    5181    5182
            5183    5221    5223    5226    5288    5291    5297    5301    5305    5312    5317    5322    5327    5332    5339
            5344    5349    5354    5360    5380    5412    5418    5434
.PAGE        812     845    4905    4948    4981    5029    5069    5109    5142
.REM           1
.REPT        792
.SBTTL      1114    1178    1229    1259    1464    1516    1543    1652    1671    1849    2024    2111    2151    2206    2256
            2363    2416    2468    2518    2565    2640    2885    3120    3310    3509    3709    3777    3827    3879    3921
            4177    4282    4361    4865    4905    4948    4981    5029    5069    5109    5142
.TITLE       775


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DZRSEC.SEQ/SOL/CRF/PAGNUM/NL:TOC=DZRSCE.SML,DZRSEC.P11
 RUN-TIME: 30 49 7 SECONDS
 RUN-TIME RATIO: 125/88=1.4
 CORE USED:  23K  (45 PAGES)
```

Spooler runtime 19 Seconds, 82 KCS, 551 disk reads, 0 disk writes, 149 pages