RP11-C/RP03

DISKLESS

MD-11-DZRPA-D

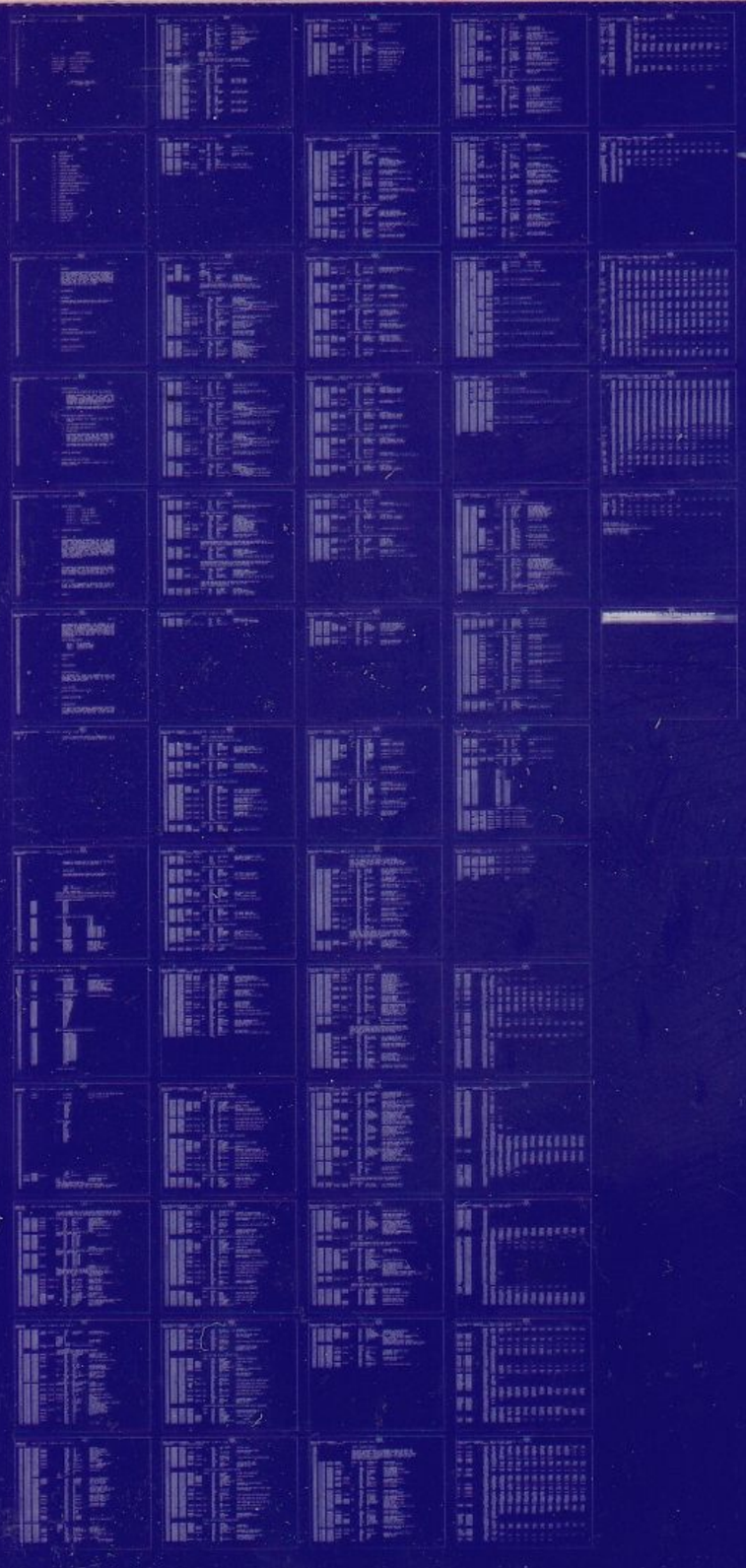
EP DZRPA-D-DL A

OCT 1976

digital

FICHE 1 OF 1

Made in U.S.A.



.REM '.

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRPA-D-D

PRODUCT NAME: RP11C DISKLESS DIAGNOSTIC

DATE CREATED: JUNE 15, 1973

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: JOE STUBBLEBINE

CONTENTS

1.0 ABSTRACT

THE RP11C DISKLESS DIAGNOSTIC EXERCISES THE RP11C IN THE MAINTENANCE MODE. IT CONSISTS OF TWO SEGMENTS. THE FIRST SEGMENT VERIFIES THE LOGIC CONTAINED IN THE RP11C BY UTILIZING THE THREE MAINTENANCE REGISTERS WHICH SIMULATE THE SIGNALS PASSING BETWEEN THE RP11C AND THE RP03. SEGMENT TWO OPERATES IN NORMAL MODE AND IS USED TO VERIFY THE SWITCHES CONTAINED ON THE RP11C AND THE RP03.

2.0 REQUIREMENTS

2.1 EQUIPMENT

STANDARD PDP-11 CONFIGURATION RP11C DISK CONTROLLER.
ONE OR MORE RP03 DISK DRIVES (SWITCH TEST ONLY).

2.2 STORAGE

PROGRAM REQUIRES 4K OF STORAGE.

2.3 PRELIMINARY PROGRAMS

NONE.

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPE.

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1.1.

4.2 STARTING ADDRESS

THE PROGRAM MAY BE STARTED AT ONE OF TWO LOCATIONS.

1. STARTING ADDRESS 200 RUNS THE DISKLESS DIAGNOSTIC. THE RP11C MUST HAVE THE MAINTENANCE SWITCH ENABLED. IF A DRIVE IS ON THE SYSTEM, IT MUST BE POWERED DOWN AND CIRCUIT BREAKER #1 LOCATED TO THE RIGHT OF THE DRIVE'S "SIGNAL IN" CABLE MUST BE TURNED OFF.

2. STARTING ADDRESS 250 RUNS THE RP11C/RP03 SWITCH TEST. AT LEAST ONE DRIVE MUST BE ON LINE.

4.3 PROGRAMS AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY USING THE ABS LOADER.

2. LOAD DESIRED STARTING ADDRESS.

3. SET SWITCHES (SEE SECTION 5.1.1).

4. PRESS START.

5. IF RUNNING THE DISKLESS TEST THE PROGRAM WILL MAKE 100 PASSES THRU THE TEST AND THEN TYPE "END OF PASS 1". THE PROGRAM WILL UPDATE THE PASS COUNT AND CONTINUE LOOPING. AFTER EACH 100 PASSES THE PASSCOUNT WILL BE TYPED OUT.

6. IF RUNNING THE SWITCH TEST, THE PROGRAM TYPES TEST COMPLETE WHEN FINISHED AND THEN HALTS.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AFTER LOADING THE STARTING ADDRESS, SELECT THE DESIRED SWITCHES.

5.1.1 SWITCH SETTINGS ARE:

SW<15>=1 ..... HALT ON ERROR

SW<14>=1 ..... LOOP ON ERROR

SW<13>=1 ..... INHIBIT PRINTOUT

SW<12> ..... NOT USED

SW<11>=1 ..... RING BELL ON ERROR

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED AT THE END OF EACH SUBTEST AND PROVIDES THE ABILITY TO LOOP ON AN ERROR. WHENEVER AN ERROR IS DETECTED, AN ERROR FLAG IS SET. THIS FLAG IS TESTED BY THE SCOPE ROUTINE. IF SET, AND LOOP ON ERROR SW<14> IS SET, THE PROGRAM WILL LOOP BACK AND REPEAT THE CONDITIONS CAUSING THE ERROR. PRIOR TO EACH SCOPE CALL THE LOOP ADDRESS IS MOVED INTO LOCATION LAD. ONCE THE PROGRAM STARTS LOOPING ON AN ERROR, IT WILL CONTINUE LOOPING EVEN THOUGH THE ERROR MAY BE INTERMITTENT. TO GO OUT OF THE LOOP RESET SW<14>.

5.2.2 HLT

THIS ROUTINE IS ENTERED UPON DETECTION OF AN ERROR. IT WILL TYPE THE PC OF THE ERROR AND ADDITIONAL ERROR INFORMATION. THIS ROUTINE TEST FOR HALT ON ERROR, INHIBIT TYPEOUTS, AND RING THE BELL. IT ALSO SETS THE ERROR FLAG USED BY THE SCOPE ROUTINE.

5.2.3 TRAP CATCHER

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

6.0 ERRORS

6.1 WHEN ERRORS ARE ENCOUNTERED, THE ADDRESS OF THE ERROR ALONG WITH THE CONTENTS OF RPDS, RPER AND RPCS ARE TYPED. BY REFERRING TO THE LISTING ADDITIONAL INFORMATION CAN BE FOUND REGARDING THE CAUSE OF THE ERROR IN THE COMMENTS. WHEN APPROPRIATE, ADDITIONAL INFORMATION IS TYPED OUT SUCH AS EXPECTED AND RECEIVED RESULTS OF AN OPERATION. ALL INFORMATION IS IN OCTAL.

ERROR MESSAGE FORMAT

```
PC =    PC OF FAILURE
RPDS =  CONTENTS OF RPDS
RPER =  CONTENTS OF RPER
RPCS =  CONTENTS OF RPCS
```

7.0 RESTRICTIONS

NONE.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

THE PROGRAM WILL MAKE 100 ITERATIONS THRU THE DISKLESS TEST TO COMPLETE ONE PASS. AT THE END OF EACH PASS, THE PASS COUNT IS TYPED OUT AND THE PROGRAM CONTINUES LOOPING.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500.

9.0 PROGRAM DESCRIPTION

9.1 DISKLESS TEST

THIS TEST UTILIZES THE RP11C MAINTENANCE REGISTERS TO CHECK THE LOGICAL RESPONSES OF THE RP11C. THE MAINTENANCE REGISTERS SIMULATE THE SIGNALS PASSING BETWEEN THE RP11C AND THE RP03. TO RUN THIS TEST

```
  299                                   THE RP11C MAINTENANCE SWITCH MUST BE ENABLED.  IF  A
  300                                   DRIVE  EXISTS,  IT  MUST BE POWERED DOWN AND CIRCUIT
```

```
                                                                    PAGE 7


                        BREAKER #1 LOCATED JUST TO THE RIGHT OF THE DRIVES'S
                        "SIGNAL IN" CABLE MUST BE TURNED OFF.

                    9.2 SWITCH TEST

                        THIS TEST FUNCTIONALLY CHECKS THE RESPONSES  OF  THE
                        SWITCHES FOUND ON THE RP11C AND THE RP03.

                            %

                        .LIST   ME
                        .NLIST  MC,MD,CND
                        .ABS
                        .TITLE  FRONT END
                    ;COPYRIGHT 1972,1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

                    ;CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS

                    ;GENERAL REGISTER ASSIGNMENTS
000000                  R0=%0
000001                  R1=%1
000002                  R2=%2
000003                  R3=%3
000004                  R4=%4
000005                  R5=%5
000006                  SP=%6
000007                  PC=%7

                    ;STATUS REGISTER (PSW) BIT ASSIGNMENTS
000001                  C=1                     ;C BIT
000002                  V=2                     ;V BIT
000004                  Z=4                     ;Z BIT
000010                  N=10                    ;N BIT
000020                  T=20                    ;T BIT
000340                  PRI7=340                ;PRIORITY LEVEL 7
000300                  PRI6=300                ;PRIORITY LEVEL 6
000240                  PRI5=240                ;PRIORITY LEVEL 5
000200                  PRI4=200                ;PRIORITY LEVEL 4
000140                  PRI3=140                ;PRIORITY LEVEL 3
000100                  PRI2=100                ;PRIORITY LEVEL 2
000040                  PRI1=40                 ;PRIORITY LEVEL 1

                    ;VECTOR ADDRESSES
000004                  ERRVEC=4                ;ERROR VECTOR
000010                  RESVEC=10               ;RESERVED INST VECTOR
000014                  TBITVEC=14              ;T BIT VECTOR
000020                  IOTVEC=20               ;IOT TRAP VECTOR
000024                  PFVEC=24                ;POWER FAIL VECTOR
000030                  EMTVEC=30               ;EMT VECTOR
```

```
000034                          TRAPVEC=34              ;TRAP VECTOR

                        ;REGISTER ADDRESSES
177776                          PSW=177776              ;PROCESSOR STATUS REGISTER
177560                          TKS=177560              ;KEYBOARD CSR
177562                          TKB=177562              ;ADDR OF KEYBOARD BUFFER
177564                          TPS=177564              ;TELEPRINTER CSR
177566                          TPB=177566              ;TELEPRINTER BUFFER
177570                          SWR=177570              ;CONSOLE SWITCH REGISTER
177570                          DISPLAY=177570          ;CONSOLE DISPLAY REGISTER

                        ;INITIAL STACK POINTER
000500                          STKPTR=500              ;PROGRAM STACK POINTER

                        ;BIT ASSIGNMENTS
100000                          B15=100000
040000                          B14=40000
020000                          B13=20000
010000                          B12=10000
004000                          B11=4000
002000                          B10=2000
001000                          B9=1000
000400                          B8=400
000200                          B7=200
000100                          B6=100
000040                          B5=40
000020                          B4=20
000010                          B3=10
000004                          B2=4
000002                          B1=2
000001                          B0=1

                        ;MEMORY MANAGEMENT REGISTER ASSIGNMENTS

177572                          SR0=177572
172340                          KIPAR0=172340
172342                          KIPAR1=172342
172344                          KIPAR2=172344
172346                          KIPAR3=172346
172350                          KIPAR4=172350
172352                          KIPAR5=172352
172354                          KIPAR6=172354
172356                          KIPAR7=172356
172300                          KIPDR0=172300
172302                          KIPDR1=172302
172304                          KIPDR2=172304
172306                          KIPDR3=172306
172310                          KIPDR4=172310
172312                          KIPDR5=172312
172314                          KIPDR6=172314
172316                          KIPDR7=172316
000006                          RW=6
000000                          UP=00


                        ;INSTRUCTION EQUATES
```

```
                104400                          HLT=TRAP                ;HLT IS A TRAP TO THE ERROR ROUTINE

                104000                          SCOPE=EMT               ;SCOPE IS AN EMT TRAP


                                        ;INDEX OF MACROS
                                        ;       .SCOPE
                                        ;       .SAVE
                                        ;       .REST
                                        ;       .ERROR
                                        ;       .PRINT
                                        ;       .DUMP
                                        ;       .RAND
                                        ;       .READ
                                        ;       .PACK

                                        ;INDEX OF CALLS
                                        ;       SCOPE
                                        ;       SAVE
                                        ;       REST
                                        ;       HLT
                                        ;       PRINT
                                        ;       DUMP
                                        ;       DUMPF
                                        ;       SDUMP
                                        ;       SDUMPF
                                        ;       RAND
                                        ;       READ
                                        ;       PACK


                                                .LIST   ME
                000200                          .=200
000200  012707  002336                          MOV     #START,PC       ;GO TO START OF TEST
                001000                          .=1000
001000  000000                          ICNT:  0                       ;CONTAINS PASS COUNT
001002  000000                          LAD:   0                       ;PROGRAM TRACE
                                        ;SCOPE (EMT) SERVICE ROUTINE
                                        ;THIS ROUTINE WILL LOOP IF AN ERROR OCCURED AND
                                        ;LOOP ON ERROR SWITCH IS SET (BIT 14). IF LOOPING IS INDICATED
                                        ;THE CONTENTS OF "LAD" EQUAL THE LOOP ADDRESS. IN ORDER
```

```
                                        ;TO LOOP ON ERROR, BIT 14 OF THE SWITCH REGISTER MUST BE SET AND
                                        ;LOCATION "ERRFLG" MUST BE NEGATIVE INDICATING AN ERROR. ONCE THE
                                        ;LOOP IS INITIATED IT WILL CONTINUE UNTIL SWITCH 14 IS CLEARED.

001004  032737  040000  177570  SCOPE$: BIT     #B14,@#SWR      ;LOOP ON ERROR?
001012  001403                          BEQ     2$              ;BRANCH IF NO
001014  005767  000220                  TST     ERRFLG          ;IS THERE AN ERROR?
001020  001003                          BNE     1$              ;BRANCH IF YES
001022  005067  000212          2$:     CLR     ERRFLG          ;RESET ERROR CONDITION
001026  000002                          RTI                     ;EXIT
001030  016716  177746          1$:     MOV     LAD,(SP)        ;MODIFY RETURN ADDRESS
001034  000002                          RTI                     ;EXIT
                                        ;ROUTINE TO SAVE REGISTERS ON THE STACK.
                                        ;CALLED BY SAVE MACRO
001036  012667  000020          SAVE$:  MOV     (SP)+,1$                ;SAVE RETURN PC
001042  010546                          MOV     R5,-(SP)
001044  010446                          MOV     R4,-(SP)
001046  010346                          MOV     R3,-(SP)
001050  010246                          MOV     R2,-(SP)
001052  010146                          MOV     R1,-(SP)
001054  010046                          MOV     R0,-(SP)
001056  016707  000000                  MOV     1$,PC           ;RETURN
001062  000000          1$:     0                               ;CONTAINS RETURN ADDRESS
                                        ;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
                                        ;CALLED BY REST MACRO
001064  012667  000020          REST$:  MOV     (SP)+,1$        ;SAVE RETURN PC
001070  012600                          MOV     (SP)+,R0
001072  012601                          MOV     (SP)+,R1
001074  012602                          MOV     (SP)+,R2
001076  012603                          MOV     (SP)+,R3
001100  012604                          MOV     (SP)+,R4
001102  012605                          MOV     (SP)+,R5
001104  016707  000000                  MOV     1$,PC           ;RETURN
001110  000000          1$:     0                               ;CONTAINS RETURN ADDR
                                        ;ERROR SERVICE ROUTINE CALLED BY HLT
                                        ;THIS ROUTINE WILL HALT ON ERROR, RING THE BELL, AND
                                        ;TRANSFER CONTROL TO A USER SUPPLIED ROUTINE IF SPECIFIED
001112  005737  177570          ERROR:  TST     @#SWR           ;HALT ON ERROR?
001116  100001                          BPL     3$              ;BRANCH IF NO
001120  000000                          HALT
001122  032737  004000  177570  3$:     BIT     #B11,@#SWR      ;RING THE BELL?
001130  001403                          BEQ     1$              ;BRANCH IF NO
001132  004567  000144                  JSR     R5,PRNTF$       ;FORCE PRINT THE MESSAGE
001136  001250                          BELL
001140  032737  020000  177570  1$:     BIT     #B13,@#SWR      ;SKIP TYPEOUT?
001146  001022                          BNE     2$              ;BRANCH IF YES
001150  004567  000110                  JSR     R5,PRINT$       ;PRINT MESSAGE
001154  001252                          ERRPC
001156  011667  000062                  MOV     (6),HLTAD$      ;GET ERROR PC+2
001162  162767  000002  000054          SUB     #2,HLTAD$       ;MODIFY
001170  117767  000050  000044          MOVB    @HLTAD$,HLTCT$  ;SAVE HLT ARGUEMENT
001176  016767  000042  000356          MOV     H TAD$,TTY
001204  004767  000134                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
001210  004767  012100                  JSR     PC,MSG          ;GO TO USER ERROR ROUTINE
001214  005737  177570          2$:     TST     @#SWR           ;HALT ON ERROR?
001220  100001                          BPL     4$              ;BRANCH IF NO
```

```
                                                HALT
001224  052767  100000  000006  4$:     BIS     #B15,ERRFLG     ;SET ERROR FLAG
001232  005267  000010                  INC     ERRORS          ;UPDATE ERROR COUNTER
001236  000002                          RTI
001240  000000                  ERRFLG: 0
001242  000000                  HLTCT$: 0
001244  000000                  HLTAD$: 0                       ;PC OF ERROR
001246  000000                  ERRORS: 0                       ;ERROR COUNT
001250  000007                  BELL:   .ASCIZ  <7>
001252  005015  005015  041520  ERRPC:  .ASCIZ  <15><12><15><12>'PC= '
001260  020075    000
        001264                          .EVEN
                                ;THIS ROUTINE WILL PRINT AN ASCIZ MESSAGE.
                                ;THE MESSAGE MUST TERMINATE IN 0
001264  032737  020000  177570  PRINT$: BIT     #B13,@#SWR      ;INHIBIT TYPEOUTS?
001272  001403                          BEQ     PRNTF$          ;BRANCH IF NO
001274  062705  000002                  ADD     #2,R5           ;UPDATE RETURN ADDR
001300  000205                          RTS     R5
001302  105737  177564          PRNTF$: TSTB    @#TPS           ;WAIT FOR PRINTER TO FINISH
001306  100375                          BPL     .-4
001310  010546                          MOV     R5,-(SP)
001312  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN PC
001316  011505                          MOV     (R5),R5         ;GET MESSAGE ADDR
001320  105715                  1$:     TSTB    (R5)            ;CHECK FOR TERMINATOR
001322  001002                          BNE     2$
001324  012605                          MOV     (SP)+,R5        ;GET RETURN ADDR
001326  000205                          RTS     R5              ;RETURN
001330  112537  177566          2$:     MOVB    (R5)+,@#TPB     ;PRINT CHARACTER
001334  105737  177564                  TSTB    @#TPS           ;WAIT TILL DONE
001340  100375                          BPL     .-4
001342  000766                          BR      1$
                                ;THIS ROUTINE TYPES A LOCATION IN OCTAL
001344  032737  020000  177570  PRINTR: BIT     #B13,@#SWR      ;INHIBIT TYPEOUT?
001352  001406                          BEQ     PRINTA          ;BRANCH IF NO
001354  000207                          RTS     PC
001356  032737  020000  177570  PRINTS: BIT     #B13,@#SWR      ;INHIBIT TYPEOUT?
001364  001405                          BEQ     PRINTB          ;BRANCH IF NO
001366  000207                          RTS     PC
001370  112767  000001  000140  PRINTA: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
001376  000402                          BR      .+6             ;SKIP
001400  005067  000132          PRINTB: CLR     .PR             ;SUPPRESS LEADING ZEROS
001404  112767  177772  000125          MOVB    #-6,.PR+1       ;SET COUNT
001412  010446                  .PTIT:  MOV     R4,-(SP)        ;SAVE R4
001414  012704  001540                  MOV     #.PR+2,R4       ;SET POINTER TO FIRST CHARACTER
001420  105014                          CLRB    (R4)            ;CLEAR FIRST BYTE
001422  000413                          BR      .PRF            ;ROTATE FIRST BIT
001424  105014                  .PRL:   CLRB    (R4)            ;CLEAR BYTE OF CHAR
001426  032767  000100  000102          BIT     #100,.PR        ;BIT TYPING MODE
001434  001006                          BNE     .PRF            ;YES SKIP 2 ROTATES
001436  006167  000120                  ROL     TTY             ;ROTATE BIT INTO C
001442  106114                          ROLB    (4)             ;PACK IT
001444  006167  000112                  ROL     TTY
001450  106114                          ROLB    (4)
001452  006167  000104          .PRF:   ROL     TTY
001456  106114                          ROLB    (4)
001460  105714                          TSTB    (4)             ;IS IT ZERO
```

```
  581  001462  001402                        BEQ     .+6             ;SKIP INC
  582  001464  105267  000046                INCB    .PR             ;SET FILL SWITCH
  583  001470  105767  000042                TSTB    .PR             ;CHECK FILL SWITCH
  584  001474  001402                        BEQ     .+6             ;SKIP BITSET
  585  001476  152724  000060                BISB    #'0,(4)+        ;MAKE INTO ASCIZ CHAR
  586  001502  105267  000031                INCB    .PR+1           ;INC COUNT
  587  001506  001346                        BNE     .PRL            ;REPEAT
  588  001510  022704  001540                CMP     #.PR+2,R4       ;EMPTY BUFFER
  589  001514  001002                        BNE     .+6             ;SKIP IF NOT
  590  001516  112724  000060                MOVB    #'0,(4)+                ;LOAD ONE ZERO
  591  001522  105014                        CLRB    (4)             ;NULL TERMINATOR
  592  001524  004567  177534                JSR     R5,PRINT$       ;PRINT MESSAGE
  593  001530  001540                        .PR+2
  594  001532  012604                        MOV     (SP)+,R4        ;RESTORE R4
  595  001534  000207                        RTS     PC
  596  001536  000012                .PR:    .BLKW   12
  597  001562  000000                TTY:    0
  598  001564                         RAND$:
  599  001564  004767  177246                JSR     PC,SAVE$        ;SAVE THE REGISTERS
  600  001570  016700  000106                MOV     LONUM,R0        ;SET R0 WITH LOW
  601  001574  016701  000100                MOV     HINUM,R1        ;SET R1 WITH HIGH
  602  001600  012703  177771                MOV     #-7,R3          ;SET SHIFT COUNT
  603  001604  005002                        CLR     R2
  604  001606  006300                1$:     ASL     R0              ;SHIFT R0 LEFT AND
  605  001610  006101                        ROL     R1              ;ROTATE CARRY INTO R1 AND
  606  001612  006102                        ROL     R2              ;ROTATE CARRY INTO R2
  607  001614  005203                        INC     R3              ;CHECK FOR DONE
  608  001616  001373                        BNE     1$
  609  001620  066702  000056                ADD     LONUM,R2        ;ADD # TO MAKE X 129
  610  001624  005501                        ADC     R1              ;PROPOGATE CARRY
  611  001626  066701  000046                ADD     HINUM,R1        ;ADD # TO MAKE X 129
  612  001632  005502                        ADC     R2              ;PROPOGATE CARRY
  613  001634  062700  001057                ADD     #1057,R0
  614  001640  005501                        ADC     R1              ;PROPOGATE CARRY
  615  001642  005502                        ADC     R2              ;PROPOGATE CARRY
  616  001644  062701  047401                ADD     #47401,R1
  617  001650  005502                        ADC     R2
  618  001652  062702  000006                ADD     #6,R2
  619  001656  060200                        ADD     R2,R0
  620  001660  005501                        ADC     R1
  621  001662  010067  000014                MOV     R0,LONUM
  622  001666  010167  000006                MOV     R1,HINUM
  623  001672  004767  177166                JSR     PC,REST$        ;RESTORE THE REGISTERS
  624  001676  000207                        RTS     PC
  625
  626  001700  000000                HINUM:  0
  627  001702  000000                LONUM:  0
  628  001704  010346                READ$:  MOV     R3,-(6)         ;SAVE R3
  629  001706  012703  002014        1$:     MOV     #INPUT$,R3      ;GET BUFFER ADDR
  630  001712  022703  002034        2$:     CMP     #INPUT$+20,R3   ;BUFFER FULL?
  631  001716  001412                        BEQ     4$              ;YES..TYPE ?
  632  001720  105737  177560                TSTB    @#177560                ;WAIT FOR A CHAR
  633  001724  100375                        BPL     .-4
  634  001726  113713  177562                MOVB    @#177562,(3)    ;GET CHAR
  635  001732  142713  000200                BICB    #200,(3)                ;GET RID OF JUNK
  636  001736  122713  000177                CMPB    #177,(3)                ;IS IT A RUBOUT?
```

```
 637  001742  001004                          BNE     3$              ;SKIP IF NO
 638  001744                          4$:
 639  001744  004567  177314                  JSR     R5,PRINT$       ;PRINT MESSAGE
 640  001750  002054                          READM$
 641  001752  000755                          BR      1$              ;CLEAR BUFFER AND START OVER
 642  001754  013737  177562  177566  3$:     MOV     @#TKB,@#TPB     ;ECHO THE CHAR
 643  001762  105737  177564                  TSTB    @#TPS
 644  001766  100375                          BPL     .-4             ;WAIT FOR READY
 645  001770  122723  000015                  CMPB    #15,(3)+                ;CHECK FOR RETURN
 646  001774  001346                          BNE     2$              ;LOOP IF NOT RETURN
 647  001776  105063  177777                  CLRB    -1(3)           ;REMOVE THE RETURN
 648  002002  004567  177256                  JSR     R5,PRINT$       ;PRINT MESSAGE
 649  002006  002060                          READL$
 650  002010  012603                          MOV     (6)+,R3         ;RESTORE R3
 651  002012  000207                          RTS     PC              ;RETURN
 652
 653  002014  000020          INPUT$: .BLKW   20
 654  002054  006477  000012  READM$: .ASCIZ  'G'<15><12>
 655  002060  000012          READL$: .ASCIZ  <12>
 656
 657                          ;TAKE THE CONTENTS OF THE TTY INPUT BUFFER AND
 658                          ;PACK THEM INTO ONE WORD TO CREATE AN OCTAL NUMBER
 659
 660  002062                  PACK$:
 661  002062  004767  176750                  JSR     PC,SAVE$        ;SAVE THE REGISTERS
 662  002066  005067  000242                  CLR     NUM$
 663  002072  005000                          CLR     R0
 664  002074  105760  002014          2$:     TSTB    INPUT$(R0)
 665  002100  001402                          BEQ     1$
 666  002102  005200                          INC     R0
 667  002104  000773                          BR      2$
 668  002106  005300                  1$:     DEC     R0
 669  002110  004767  000166                  JSR     PC,PAC$         ;GET OCTAL CHAR
 670  002114  016767  000212  000212          MOV     PK$,NUM$        ;PACK FIRST CHAR
 671  002122  004767  000154                  JSR     PC,PAC$         ;GET OCTAL CHAR
 672  002126  000241                          CLC
 673  002130  006167  000176                  ROL     PK$
 674  002134  006167  000172                  ROL     PK$
 675  002140  006167  000166                  ROL     PK$
 676  002144  056767  000162  000162          BIS     PK$,NUM$        ;PACK SECOND CHAR
 677  002152  004767  000124                  JSR     PC,PAC$         ;GET OCTAL CHAR
 678  002156  000241                          CLC
 679  002160  000367  000146                  SWAB    PK$
 680  002164  006067  000142                  ROR     PK$
 681  002170  006067  000136                  ROR     PK$
 682  002174  056767  000132  000132          BIS     PK$,NUM$        ;PACK THIRD CHAR
 683  002202  004767  000074                  JSR     PC,PAC$         ;GET OCTAL CHAR
 684  002206  000367  000120                  SWAB    PK$
 685  002212  000241                          CLC
 686  002214  006167  000112                  ROL     PK$
 687  002220  056767  000106  000106          BIS     PK$,NUM$        ;PACK FOURTH CHAR
 688  002226  004767  000050                  JSR     PC,PAC$         ;GET OCTAL CHAR
 689  002232  000367  000074                  SWAB    PK$
 690  002236  000241                          CLC
 691  002240  006167  000066                  ROL     PK$
 692  002244  006167  000062                  ROL     PK$
```

```
693  002250  006167  000056                  ROL     PK$
694  002254  006167  000052                  ROL     PK$
695  002260  056767  000046  000046          BIS     PK$,NUM$        ;PACK FIFTH CHAR
696  002266  000402                          BR      PKEX1$
697  002270  062706  000002          PKEX$:  ADD     #2,SP           ;MODIFY STACK
698  002274                          PKEX1$:
699  002274  004767  176564                  JSR     PC,REST$        ;RESTORE THE REGISTERS
700  002300  000207                          RTS     PC              ;EXIT
701
702  002302  005700                  PAC$:   TST     R0
703  002304  100771                          BMI     PKEX$
704  002306  005067  000020                  CLR     PK$
705  002312  116067  002014  000012          MOVB    INPUT$(R0),PK$  ;GET INPUT CHAR
706  002320  005300                          DEC     R0
707  002322  042767  177770  000002          BIC     #177770,PK$     ;CLEAR UNWANTED BITS
708  002330  000207                          RTS     PC
709
710  002332  000000                  PK$:    0
711  002334  000000                  NUM$:   0
```

```
                                        .TITLE  RP11C DISKLESS DIAGNOSTIC
            002336                      BACK=.
            000250                      .=250
000250      012707  011102                      MOV     #SWTST,PC
            002336                      .=BACK
            000254                      VECTOR=254
            000256                      STATUS=256
002336      012706  000500              START:  MOV     #STKPTR,SP      ;SETUP STACK
002342      005067  011250                      CLR     PASSCT          ;CLEAR THE PASSCOUNT
002346      005067  176426              LOOP:   CLR     ICNT            ;CLEAR THE ITERATION COUNT
002352      004767  010676                      JSR     PC,INIT         ;INITIALIZE VECTORS

                                        ;TEST LOADING AND READING OF ALL POSSIBLE BITS IN RP11C
                                        ;REGISTERS USING A PATTERN OF WALKING 0'S (-2,-3,-5 ETC) AND
                                        ;WALKING 1'S (1,2,4,10 ETC) ALSO TEST RESET ON EACH REGISTER


                                        ;RPER-ERROR REGISTER

002356                                  ERBT:
002356      004567  010430                      JSR     R5,BITST        ;TEST BITS IN REGISTER
002362      176656                              .WORD   176656          ;MASK=176656
002364      176712                              .WORD   176712          ;REGISTER=RPER
002366      104403                              HLT     +3              ;INCORRECT DATA RECEIVED FROM RPER
002370      000207                              RTS     PC              ;EXIT FROM ERROR CALL
002372      012737  176656  176712  2$:         MOV     #176656,@#176712                ;SET MASK BITS IN THE REGISTER
002400      000005                              RESET                   ;CLEAR REGISTER RPER
002402      005737  176712                      TST     @#176712                ;DID RESET CLEAR THE REGISTER
002406      001406                              BEQ     1$              ;BRANCH IF YES
002410      013767  176712  011070              MOV     @#176712,REC$
002416      005067  011062                      CLR     EXP$
002422      104403                              HLT     +3              ;REGISTER RPER DID NOT CLEAR WITH RESET
002424      012767  002372  176350  1$:         MOV     #2$,LAD         ;SET UP ERROR LOOP ADDR
002432      104000                              SCOPE
002434      012777  176656  011170  ERGO:   MOV     #176656,@RPER   ;SET ALL POSSIBLE BITS IN RPER
002442      012777  000001  011164          MOV     #1,@RPCS        ;CLEAR RP11C
002450      017767  011156  011030          MOV     @RPER,REC$      ;GET CONTENTS OF RPER
002456      005767  011024                  TST     REC$            ;DID GO CLEAR RPER?
002462      001403                          BEQ     1$              ;BRANCH IF YES
002464      005067  011014                  CLR     EXP$
002470      104403                          HLT     +3              ;GO DID NOT CLEAR RPER
002472      012767  002434  176302  1$:     MOV     #ERGO,LAD       ;SETUP ERROR LOOP ADDR
002500      104000                          SCOPE                   ;LOOP ON TEST IF ERROR

                                        ;RPCS-CONTROL STATUS REGISTER

002502      004567  010304          CSBT:   JSR     R5,BITST        ;TEST BITS IN REGISTER
002506      037576                          .WORD   37576           ;MASK=37576
002510      176714                          .WORD   176714          ;REGISTER=RPCS
002512      104403                          HLT     +3              ;INCORRECT DATA RECEIVED FROM RPCS
002514      000207                          RTS     PC              ;EXIT FROM ERROR CALL
002516      012777  037576  011110  2$:     MOV     #37576,@RPCS    ;SET MASK BITS IN REGISTER
002524      000005                          RESET                   ;CLEAR THE SYSTEM
002526      017767  011102  010752          MOV     @RPCS,REC$      ;GET CONTENTS OF RPCS
002534      042767  000200  010744          BIC     #87,REC$        ;CLEAR THE READY BIT
002542      001404                          BEQ     1$              ;BRANCH IF RESULT IS ZERO
```

```
                                        CLR     EXP$
                                        HLT     +3              ;RESET DID NOT CLEAR RPCS
                                        BR      3$
002554  032777  000200  011052  1$:   BIT     #B7,@RPCS       ;IS READY SET?
002562  001001                        BNE     3$
002564  104400                        HLT                     ;READY NOT SET IN RPCS AFTER RESET
002566  012767  002516  176206  3$:   MOV     #2$,LAD         ;SETUP LOOP ADDR
002574  104000                        SCOPE

                                ;RPWC-WORD COUNT REGISTER

002576                          RPBT:
002576  004567  010210                JSR     R5,BITST        ;TEST BITS IN REGISTER
002602  177777                        .WORD   177777          ;MASK=177777
002604  176716                        .WORD   176716          ;REGISTER=RPWC
002606  104403                        HLT     +3              ;INCORRECT DATA RECEIVED FROM RPWC
002610  000207                        RTS     PC              ;EXIT FROM ERROR CALL
002612  012737  177777  176716  2$:   MOV     #177777,@#176716                ;SET MASK BITS IN THE REGISTER
002620  000005                        RESET                   ;CLEAR REGISTER RPWC
002622  005737  176716                TST     @#176716                ;DID RESET CLEAR THE REGISTER
002626  001406                        BEQ     1$              ;BRANCH IF YES
002630  013767  176716  010650        MOV     @#176716,REC$
002636  005067  010642                CLR     EXP$
002642  104403                        HLT     -3              ;REGISTER RPWC DID NOT CLEAR WITH RESET
002644  012767  002612  176130  1$:   MOV     #2$,LAD         ;SET UP ERROR LOOP ADDR
002652  104000                        SCOPE

                                ;RPBA- BUS ADDRESS REGISTER

002654                          BABT:
002654  004567  010132                JSR     R5,BITST        ;TEST BITS IN REGISTER
002660  177777                        .WORD   177777          ;MASK=177777
002662  176720                        .WORD   176720          ;REGISTER=RPBA
002664  104403                        HLT     +3              ;INCORRECT DATA RECEIVED FROM RPBA
002666  000207                        RTS     PC              ;EXIT FROM ERROR CALL
002670  012737  177777  176720  2$:   MOV     #177777,@#176720                ;SET MASK BITS IN THE REGISTER
002676  000005                        RESET                   ;CLEAR REGISTER RPBA
002700  005737  176720                TST     @#176720                ;DID RESET CLEAR THE REGISTER
002704  001406                        BEQ     1$              ;BRANCH IF YES
002706  013767  176720  010572        MOV     @#176720,REC$
002714  005067  010564                CLR     EXP$
002720  104403                        HLT     +3              ;REGISTER RPBA DID NOT CLEAR WITH RESET
002722  012767  002670  176052  1$:   MOV     #2$,LAD         ;SET UP ERROR LOOP ADDR
002730  104000                        SCOPE

                                ;RPCA-CYLINDER ADDRESS REGISTER

002732                          CABT:
002732  004567  010054                JSR     R5,BITST        ;TEST BITS IN REGISTER
002736  000777                        .WORD   777             ;MASK=777
002740  176722                        .WORD   176722          ;REGISTER=RPCA
002742  104403                        HLT     +3              ;INCORRECT DATA RECEIVED FROM RPCA
002744  000207                        RTS     PC              ;EXIT FROM ERROR CALL
002746  012737  000777  176722  2$:   MOV     #777,@#176722           ;SET MASK BITS IN THE REGISTER
002754  000005                        RESET                   ;CLEAR REGISTER RPCA
002756  005737  176722                TST     @#176722                ;DID RESET CLEAR THE REGISTER
```

```
                                        BEQ     1$                      ;BRANCH IF YES
                                        MOV     @#176722,REC$
                                        CLR     EXP$
                                        HLT     +3                      ;REGISTER RPCA DID NOT CLEAR WITH RESET
                                1$:     MOV     #2$,LAD                 ;SET UP ERROR LOOP ADDR
                                        SCOPE

                                ;RPDA-DISK ADDRESS REGISTER

                                DABT:   JSR     R5,BITST                ;TEST BITS IN REGISTER
                                        .WORD   17417                   ;MASk=17417
                                        .WORD   176724                  ;REGISTER=RPDA
                                        HLT     +3                      ;INCORRECT DATA RECEIVED FROM RPDA
                                        RTS     PC                      ;EXIT FROM ERROR CALL
                                2$:     MOV     #17417,@RPDA            ;SET MASK BITS IN RPDA
                                        RESET                           ;CLEAR THE CONTROLLER
                                        MOV     @RPDA,REC$              ;GET CONTENTS OF RPDA
                                        BIC     #360,REC$               ;CLEAR UNWANTED BITS
                                        BEQ     1$                      ;BRANCH IF RESULT IS ZERO
                                        CLR     EXP$
                                        HLT     +3                      ;RESET DID NOT CLEAR RPDA
                                1$:     MOV     #2$,LAD                 ;SETUP LOOP ADDR
                                        SCOPE

                                ;WC REGISTER RAPID DATA TEST - A FAILURE HERE MAY NOT REPORT AS A
                                ;DROP OR PICKED BIT(S) IN ERROR TYPOUT, THIS INDICATES THAT THE REGISTER
                                ;DOES NOT SETTLE WITH THE PROPER DATA FAST ENOUGH

                                RAPWC:  JSR     R5,RAPBIS
                                        .WORD   176716                  ;REGISTER = RPWC
                                        HLT     +3                      ;INCORRECT DATA FROM RPWC
                                        RTS     PC                      ;EXIT FROM ERROR CALL
                                        CLR     @#176716                ;CLEAR RPWC
                                        MOV     #RAPWC,LAD              ;IF LOOPING ON ERROR, SETUP FOR THIS TEST
                                        SCOPE

                                ;BA REGISTER RAPID DATA TEST - A FAILURE HERE MAY NOT REPORT AS A DROP
                                ;OR PICKED BIT(S) IN ERROR TYPOUT, THIS INDICATES THAT THE REGISTER
                                ;DOES NOT SETTLE WITH THE PROPER DATA FAST ENOUGH.

                                RAPBA:  JSR     R5,RAPBIS
                                        .WORD   176720                  ;REGISTER = RPBA
                                        HLT     +3                      ;INCORRECT DATA FROM RPBA
                                        RTS     PC                      ;RETURN FROM ERROR CALL
                                        CLR     @#176720                ;CLEAR RPBA
                                        MOV     #RAPBA,LAD              ;IF LOOPING ON ERROR, SETUP FOR THIS TEST
                                        SCOPE

                                ;TEST THE ABILITY OF BIT 0 OF RPCS TO CLEAR THE CONTROLLER
                                ;WHEN THE COMMAND BITS ARE 0.  ERR SYNC(1) AND
                                ;RESET C SHOULD GENERATE INIT L

                                RESWC:  MOV     #-1,@RPWC               ;SET RPWC
                                        JSR     PC,CLRP                 ;CLEAR THE RP11C
                                        MOV     @RPWC,REC$              ;GET CONTENTS OF RPWC
                                        TST     REC$                    ;DID INIT L CLEAR RPWC?
```

```
  880  003170  001403                        BEQ     1$              ;BRANCH IF YES
  881  003172  005067  010306                CLR     EXP$            ;CLEAR EXPECTED RESULTS
  882  003176  104403                        HLT     +3              ;INIT L DID NOT CLEAR RPWC
  883  003200  012767  003144  175574  1$:   MOV     #RESWC,LAD
  884  003206  104000                        SCOPE
```

```
                                        .SBTTL  ***RPDS FUNCTION TEST***

                                        ;TEST THE SETTING OF SELECTED UNIT READY

003210  005077  010444        DSF1:   CLR     @RPM3
003214  005077  010410                CLR     @RPDS
003220  052777  040000  010432        BIS     #B14,@RPM3      ;SET MAINT UNIT READY
003226  032777  100000  010374        BIT     #B15,@RPDS      ;DID SELECTED UNIT READY SET?
003234  001001                        BNE     1$              ;BRANCH IF SET
003236  104400                        HLT                     ;SELECTED UNIT READY NOT SET
003240  012767  003210  175534  1$:   MOV     #DSF1,LAD
003246  104000                        SCOPE

                                        ;TEST SELECTED UNIT READY TO CLEAR

003250  005077  010404        DSF2:   CLR     @RPM3
003254  052777  040000  010376        BIS     #B14,@RPM3      ;SET MAINT UNIT READY
003262  042777  040000  010370        BIC     #B14,@RPM3      ;CLEAR MAINT UNIT READY
003270  032777  100000  010332        BIT     #B15,@RPDS      ;DID SELECTED UNIT READY CLEAR?
003276  001401                        BEQ     1$
003300  104400                        HLT                     ;SELECTED UNIT READY DID NOT CLEAR

003302  012767  003250  175472  1$:   MOV     #DSF2,LAD
003310  104000                        SCOPE


                                        ;TEST THE SETTING OF SEEK INCOMPLETE

003312  000005                DSF4:   RESET
003314  005077  010340                CLR     @RPM3
003320  005077  010304                CLR     @RPDS
003324  052777  002000  010326        BIS     #B10,@RPM3      ;SET MAINT. SEEK INCOMPLETE
003332  032777  004000  010270        BIT     #B11,@RPDS      ;DID SEEK INCOMPLETE SET?
003340  001002                        BNE     1$
003342  104400                        HLT                     ;SEEK INCOMPLETE DID NOT SET
003344  000421                        BR      2$
003346  032777  000001  010256  1$:   BIT     #B0,@RPER       ;DID DISK ERROR SET?
003354  001002                        BNE     3$              ;BRANCH IF YES
003356  104400                        HLT                     ;DISK ERROR DID NOT SET AFTER SUSI
003360  000413                        BR      2$
003362  032777  040000  010244  3$:   BIT     #B14,@RPCS      ;DID HARD ERROR SET?
003370  001002                        BNE     4$              ;BRANCH IF SET
003372  104400                        HLT                     ;HARD ERROR DID NOT SET AFTER SUSI
003374  000405                        BR      2$
003376  032777  100000  010230  4$:   BIT     #B15,@RPCS      ;DID ERROR SET?
003404  001001                        BNE     2$              ;BRANCH IF SET
003406  104400                        HLT                     ;ERROR DID NOT SET AFTER SUSI
003410  012767  003312  175364  2$:   MOV     #DSF4,LAD
003416  104000                        SCOPE

                                        ;TEST SEEK INCOMPLETE TO CLEAR

003420  005077  010234        DSF5:   CLR     @RPM3
003424  005077  010200                CLR     @RPDS
003430  052777  002000  010222        BIS     #B10,@RPM3      ;SET MAINT SEEK INCOMPLETE
003436  042777  002000  010214        BIC     #B10,@RPM3      ;CLEAR
```

```
  941  003444  032777  004000  010156          BIT     #B11,@RPDS      ;DID SEEK INCOMPLETE CLEAR?
  942  003452  001401                          BEQ     1$              ;BRANCH IF CLEAR
  943  003454  104400                          HLT                     ;SEEK INCOMPLETE DID NOT CLEAR
  944  003456  012767  003420  175316  1$:     MOV     #DSF5,LAD
  945  003464  104000                          SCOPE
  946
  947                                  ;TEST THE SETTING OF FILE UNSAFE
  948
  949  003466  005077  010166          DSF7:   CLR     @RPM3
  950  003472  005077  010132                  CLR     @RPDS
  951  003476  052777  004000  010154          BIS     #B11,@RPM3      ;SET MAINT. FILE UNSAFE
  952  003504  032777  001000  010116          BIT     #B9,@RPDS       ;DID FILE UNSAFE SET?
  953  003512  001001                          BNE     1$
  954  003514  104400                          HLT                     ;FILE UNSAFE DID NOT SET
  955  003516  012767  003466  175256  1$:     MOV     #DSF7,LAD
  956  003524  104000                          SCOPE
  957
  958                                  ;TEST CLEARING OF FILE UNSAFE
  959
  960  003526  005077  010126          DSF10:  CLR     @RPM3
  961  003532  005077  010072                  CLR     @RPDS
  962  003536  052777  004000  010114          BIS     #B11,@RPM3      ;SET MAINT FILE UNSAFE
  963  003544  042777  004000  010106          BIC     #B11,@RPM3      ;CLEAR
  964  003552  032777  001000  010050          BIT     #B9,@RPDS       ;IS FILE UNSAFE CLEAR?
  965  003560  001401                          BEQ     1$
  966  003562  104400                          HLT                     ;FILE UNSAFE DID NOT CLEAR
  967  003564  012767  003526  175210  1$:     MOV     #DSF10,LAD
  968  003572  104000                          SCOPE
  969
  970                                  ;TEST THE SETTING OF WRITE PROTECT
  971
  972  003574  005077  010060          DSF11:  CLR     @RPM3
  973  003600  005077  010024                  CLR     @RPDS
  974  003604  052777  100000  010046          BIS     #B15,@RPM3      ;SET MAINT READ ONLY
  975  003612  032777  000400  010010          BIT     #B8,@RPDS       ;DID WIRTE PROTECT SET?
  976  003620  001001                          BNE     1$
  977  003622  104400                          HLT                     ;WROTE PROTECT DID NOT SET
  978  003624  012767  003574  175150  1$:     MOV     #DSF11,LAD
  979  003632  104000                          SCOPE
  980
  981                                  ;TEST THE CLEARING OF WRITE PROTECT
  982
  983  003634  005077  010020          DSF12:  CLR     @RPM3
  984  003640  005077  007764                  CLR     @RPDS
  985  003644  052777  100000  010006          BIS     #B15,@RPM3      ;SET MAINT READ ONLY
  986  003652  042777  100000  010000          BIC     #B15,@RPM3      ;CLEAR
  987  003660  032777  000400  007742          BIT     #B8,@RPDS       ;DID WRITE PROTECT CLEAR?
  988  003666  001401                          BEQ     1$
  989  003670  104400                          HLT                     ;WRITE PROTECT DID NOT CLEAR
  990  003672  012767  003634  175102  1$:     MOV     #DSF12,LAD
  991  003700  104000                          SCOPE
  992
  993                                  ;TEST SETTING OF ATTENTION BITS
  994
  995  003702  012767  000001  007574  DSF13:  MOV     #1,EXP$         ;INITIATE ATTENTION BIT PATTERN
  996  003710  005077  007740          2$:     CLR     @RPM2
```

```
  997  003714  005077  007740                    CLR     @RPM3
  998  003720  012777  000377  007702            MOV     #377,@RPDS      ;CLEAR ATTENTION BITS
  999  003726  016777  007552  007720            MOV     EXP$,@RPM2      ;SET MAINT ATTENTION BIT
 1000  003734  126777  007544  007666            CMPB    EXP$,@RPDS      ;DID THE ATTEN BIT SET IN RPDS?
 1001  003742  001404                            BEQ     1$              ;BRANCH IF OK
 1002  003744  117767  007660  007534            MOVB    @RPDS,REC$
 1003  003752  104401                            HLT     +1              ;ATTENTION BIT DID NOT SET PROPERLY
 1004  003754  012767  003710  175020  1$:       MOV     #2$,LAD
 1005  003762  104000                            SCOPE
 1006  003764  016777  007514  007636  4$:       MOV     EXP$,@RPDS      ;CLEAR ATTENTION BIT
 1007  003772  105777  007632                    TSTB    @RPDS           ;DID IT CLEAR?
 1008  003776  001401                            BEQ     3$              ;BRANCH IF CLEAR
 1009  004000  104400                            HLT                     ;ATTENTION BIT DID NOT CLEAR
 1010  004002  012767  003764  174772  3$:       MOV     #4$,LAD
 1011  004010  104000                            SCOPE
 1012  004012  000241                            CLC
 1013  004014  006167  007464                    ROL     EXP$            ;ROTATE PATTERN
 1014  004020  032767  000400  007456            BIT     #88,EXP$        ;END OF PATTERN?
 1015  004026  001730                            BEQ     2$              ;BRANCH IF NO
 1016  004030  012777  000377  007616  6$:       MOV     #377,@RPM2      ;SET ATTENTION BITS
 1017  004036  000005                            RESET
 1018  004040  105777  007564                    TSTB    @RPDS           ;DID RESET CLEAR ATTN BITS?
 1019  004044  001401                            BEQ     5$
 1020  004046  104400                            HLT                     ;RESET DID NOT CLEAR ATTENTION BITS
 1021  004050  012767  004030  174724  5$:       MOV     #6$,LAD
 1022  004056  104000                            SCOPE
 1023  004060  005077  007570          8$:       CLR     @RPM2
 1024  004064  012777  000377  007562            MOV     #377,@RPM2      ;SET ALL ATTENTION BITS
 1025  004072  005077  007532                    CLR     @RPDS           ;ISSUE CLEAR RPDS
 1026  004076  122777  000377  007524            CMPB    #377,@RPDS      ;DID ATTENTION BITS REMAIN SET?
 1027  004104  001410                            BEQ     7$              ;BRANCH IF YES
 1028  004106  005067  007372                    CLR     EXP$
 1029  004112  005067  007370                    CLR     REC$
 1030  004116  117767  007506  007362            MOVB    @RPDS,REC$      ;GET BAD DATA
 1031  004124  104403                            HLT     +3              ;ATTENTION BITS CLEARED WITH A ZERO
 1032  004126  012767  004060  174646  7$:       MOV     #8$,LAD
 1033  004134  104000                            SCOPE
```

```
1034                                        .SBTTL  ***RPER FUNCTION TEST***
1035
1036                                        ;TEST THE SETTING OF WRITE PROTECT VIOLATION
1037
1038  004136  000005                ERF1:   RESET
1039  004140  052777  100000  007512        BIS     #B15,@RPM3      ;SET MAINT READ ONLY
1040  004146  012777  177777  007464        MOV     #-1,@RPWC
1041  004154  112777  000003  007452        MOVB    #3,@RPCS        ;ISSUE A WRITE
1042  004162  012767  160001  007434        MOV     #B15+B14+B13+B0,TPL
1043  004170  004767  007366                JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
1044  004174  032777  100000  007430        BIT     #B15,@RPER      ;DID WRITE VIOLATION SET?
1045  004202  001002                        BNE     1$
1046  004204  104400                        HLT                     ;WRITE VIOLATION DID NOT SET
1047  004206  000413                        BR      2$
1048
1049  004210  032777  040000  007416  1$:   BIT     #B14,@RPCS      ;DID HARD ERROR SET AFTER WPV?
1050  004216  001002                        BNE     3$
1051  004220  104400                        HLT                     ;HARD ERROR DID NOT SET AFTER WPV
1052  004222  000405                        BR      2$
1053  004224  032777  100000  007402  3$:   BIT     #B15,@RPCS      ;DID ERROR SET AFTER WPV?
1054  004232  001001                        BNE     2$
1055  004234  104400                        HLT                     ;ERROR DID NOT SET AFTER WPV
1056  004236  012767  004136  174536  2$:   MOV     #ERF1,LAD
1057  004244  104000                        SCOPE
1058
1059                                        ;TEST THE SETTING OF FILE UNSAFE VIOLATION
1060
1061  004246  000005                ERF2:   RESET
1062  004250  005077  007356                CLR     @RPER
1063  004254  052777  004000  007376        BIS     #B11,@RPM3      ;SET MAINT FILE UNSAFE
1064  004262  012777  177777  007350        MOV     #-1,@RPWC
1065  004270  112777  000003  007336        MOVB    #3,@RPCS        ;ISSUE WRITE
1066  004276  012767  064001  007320        MOV     #B14+B13+B11+B0,TPL
1067  004304  004767  007252                JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
1068  004310  032777  040000  007314        BIT     #B14,@RPER      ;DID FILE UNSAFE VIOLATION SET?
1069  004316  001002                        BNE     1$
1070  004320  104400                        HLT                     ;FILE UNSAFE VIOLATION DID NOT SET
1071  004322  000413                        BR      2$
1072  004324  032777  040000  007302  1$:   BIT     #B14,@RPCS      ;DID HARD ERROR SET AFTER FUV
1073  004332  001002                        BNE     3$
1074  004334  104400                        HLT                     ;HARD ERROR DID NOT SET AFTER FUV
1075  004336  000405                        BR      2$
1076  004340  032777  100000  007266  3$:   BIT     #B15,@RPCS      ;DID ERROR SET
1077  004346  001001                        BNE     2$
1078  004350  104400                        HLT                     ;ERROR DID NOT SET
1079  004352  012767  004246  174422  2$:   MOV     #ERF2,LAD
1080  004360  104000                        SCOPE
1081
1082                                        ;NON-EXISTENT CYLINDER SHOULD NOT SET WITH LEGAL ADDRESSES
1083
1084  004362  005067  007116        ERF3:   CLR     EXP$            ;START AT ADDR 0
1085  004366  004767  007116        2$:     JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1086  004372  005077  007234                CLR     @RPER
1087  004376  016777  007102  007240        MOV     EXP$,@RPCA      ;LOAD CYLINDER ADDR
1088  004404  012777  177777  007226        MOV     #-1,@RPWC
1089  004412  012777  000003  007214        MOV     #3,@RPCS        ;ISSUE A WRITE
```

```
 1090  004420  012767  060001  007176          MOV     #B14+B13+B0,TPL
 1091  004426  004767  007130                  JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
 1092  004432  032777  020000  007172          BIT     #B13,@RPER      ;DID NON-EXISTENT CYLINDER SET?
 1093  004440  001411                          BEQ     1$
 1094  004442  104400                          HLT                     ;NON-EXISTENT CYLINDER SET ON LEGAL ADDR
 1095  004444  004567  174614                  JSR     R5,PRINT$       ;PRINT MESSAGE
 1096  004450  013757                          MES10
 1097  004452  016767  007036  175102          MOV     EXP$,TTY
 1098  004460  004767  174660                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
 1099  004464  012767  004366  174310  1$:     MOV     #2$,LAD
 1100  004472  104000                          SCOPE
 1101  004474  005267  007004                  INC     EXP$            ;UPDATE CYLINDER ADDR
 1102  004500  022767  000626  006776          CMP     #626,EXP$       ;IS ADDR STILL LEGAL?
 1103  004506  001327                          BNE     2$              ;BRANCH IF YES
 1104
 1105                                  ;TEST SETTING OF NON-EXISTENT CYLINDER WITH ILLEGAL CYL ADDR
 1106
 1107  004510  004767  006774          ERF4:   JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 1108  004514  005077  007112                  CLR     @RPER
 1109  004520  016777  006760  007116          MOV     EXP$,@RPCA      ;LOAD CYLINDER ADDR
 1110  004526  012777  177777  007104          MOV     #-1,@RPWC
 1111  004534  012777  000003  007072          MOV     #3,@RPCS        ;ISSUE WRITE
 1112  004542  012767  060001  007054          MOV     #B14+B13+B0,TPL
 1113  004550  004767  007006                  JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
 1114  004554  032777  020000  007050          BIT     #B13,@RPER      ;DID NON-EXISTENT CYL SET?
 1115  004562  001012                          BNE     1$
 1116  004564  104400                          HLT                     ;NON-EXISTENT CYL DID NOT SET
 1117  004566  004567  174472                  JSR     R5,PRINT$       ;PRINT MESSAGE
 1118  004572  013757                          MES10
 1119  004574  016767  006704  174760          MOV     EXP$,TTY
 1120  004602  004767  174536                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
 1121  004606  000413                          BR      2$
 1122  004610  032777  040000  007016  1$:     BIT     #B14,@RPCS      ;DID HARD ERROR SET AFTER NXC?
 1123  004616  001002                          BNE     3$
 1124  004620  104400                          HLT                     ;HARD ERROR DID NOT SET
 1125  004622  000405                          BR      2$
 1126  004624  032777  100000  007002  3$:     BIT     #B15,@RPCS      ;DID ERROR SET AFTER NXC
 1127  004632  001001                          BNE     2$
 1128  004634  104400                          HLT                     ;ERROR DID NOT SET
 1129  004636  012767  004510  174136  2$:     MOV     #ERF4,LAD
 1130  004644  104000                          SCOPE
 1131  004646  005267  006632                  INC     EXP$            ;UPDATE CYLINDER ADDR
 1132  004652  022767  000700  006624          CMP     #700,EXP$       ;PATTERN EXCEEDED?
 1133  004660  001313                          BNE     ERF4
 1134
 1135                                  ;NON-EXISTENT TRACK SHOULD NOT SET WITH LEGAL ADDRESSES
 1136
 1137  004662  000005                  ERF5:   RESET
 1138  004664  005067  006614                  CLR     EXP$            ;STARTING TRACK ADDR OF 0
 1139  004670  005077  006752                  CLR     @RPDA
 1140  004674  004767  006610          ERL5:   JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 1141  004700  005077  006726                  CLR     @RPER
 1142  004704  116777  006574  006736          MOVB    EXP$,@RPDA1     ;LOAD TRACK ADDRESS
 1143  004712  012777  177777  006720          MOV     #-1,@RPWC
 1144  004720  012777  000003  006706          MOV     #3,@RPCS        ;WRITE
 1145  004726  012767  060001  006670          MOV     #B14+B13+B0,TPL
```

```
1146  004734  004767  006622              JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
1147  004740  032777  010000  006664      BIT     #B12,@RPER      ;IS NXT SET?
1148  004746  001411                      BEQ     1$
1149  004750  104400                      HLT                     ;NXT SET ON LEGAL ADDR
1150  004752  004567  174306              JSR     R5,PRINT$       ;PRINT MESSAGE
1151  004756  013757                      MES10
1152  004760  016767  006520  174574      MOV     EXP$,TTY
1153  004766  004767  174352              JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
1154  004772  012767  004674  174002  1$: MOV     #ERL5,LAD
1155  005000  104000                      SCOPE
1156  005002  005267  006476              INC     EXP$            ;INCREMENT TRACK ADDR
1157  005006  022767  000024  006470      CMP     #24,EXP$        ;IS ADDR STILL LEGAL?
1158  005014  001327                      BNE     ERL5            ;BRANCH IF YES
1159
1160
1161                                  ;TEST SETTING OF NON-EXISTENT TRACK
1162
1163  005016  004767  006466          ERF6: JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1164  005022  005077  006604              CLR     @RPER
1165  005026  116777  006452  006614      MOVB    EXP$,@RPDA1     ;LOAD TRACK ADDR
1166  005034  012777  177777  006576      MOV     #-1,@RPWC
1167  005042  012777  000003  006564      MOV     #3,@RPCS        ;WRITE
1168  005050  012767  060001  006546      MOV     #B14+B13+B0,TPL
1169  005056  004767  006500              JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
1170  005062  032777  010000  006542      BIT     #B12,@RPER      ;DID NXT SET?
1171  005070  001012                      BNE     1$
1172  005072  104400                      HLT                     ;NXT DID NOT SET
1173  005074  004567  174164              JSR     R5,PRINT$       ;PRINT MESSAGE
1174  005100  013757                      MES10
1175  005102  016767  006376  174452      MOV     EXP$,TTY
1176  005110  004767  174230              JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
1177  005114  000413                      BR      2$
1178  005116  032777  040000  006510  1$: BIT     #B14,@RPCS      ;DID HARD ERROR SET AFTER NXT?
1179  005124  001002                      BNE     3$
1180  005126  104400                      HLT                     ;HARD ERROR DID NOT SET AFTER NXT
1181  005130  000405                      BR      2$
1182  005132  032777  100000  006474  3$: BIT     #B15,@RPCS      ;DID ERROR SET AFTER NXT?
1183  005140  001001                      BNE     2$
1184  005142  104400                      HLT                     ;ERROR DID NOT SET AFTER NXT
1185  005144  012767  005016  173630  2$: MOV     #ERF6,LAD
1186  005152  104000                      SCOPE
1187  005154  005267  006324              INC     EXP$            ;INCREMENT TRACK ADDR.
1188  005160  022767  000040  006316      CMP     #40,EXP$        ;END OF PATTERN?
1189  005166  001313                      BNE     ERF6            ;BRANCH IF NO
1190
1191                                  ;NON-EXISTENT SECTOR SHOULD NOT SET WITH LEGAL SECTOR ADDRESSES
1192
1193  005170  005077  006452          ERF7: CLR     @RPDA
1194  005174  005067  006304              CLR     EXP$            ;STARTING SECTOR ADDR OF 0
1195  005200  004767  006304          ERL7: JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1196  005204  005077  006422              CLR     @RPER
1197  005210  016777  006270  006430      MOV     EXP$,@RPDA      ;LOAD SECTOR ADDR
1198  005216  012777  177777  006414      MOV     #-1,@RPWC
1199  005224  012777  000003  006402      MOV     #3,@RPCS                ;WRITE
1200  005232  012767  060001  006364      MOV     #B14+B13+B0,TPL
1201  005240  004767  006316              JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
```

```
1202  005244  032777  004000  006360          BIT     #B11,@RPER      ;DID NXS SET?
1203  005252  001411                          BEQ     1$
1204  005254  104400                          HLT                     ;NXS SET ON LEGAL ADDR
1205  005256  004567  174002                  JSR     R5,PRINT$       ;PRINT MESSAGE
1206  005262  013757                          MES10
1207  005264  016767  006214  174270          MOV     EXP$,TTY
1208  005272  004767  174046                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
1209  005276  012767  005200  173476  1$:     MOV     #ERL7,LAD
1210  005304  104000                          SCOPE
1211  005306  005267  006172                  INC     EXP$            ;UPDATE SECTOR ADDR
1212  005312  022767  000012  006164          CMP     #12,EXP$        ;IS ADDR STILL LEGAL?
1213  005320  001327                          BNE     ERL7            ;BRANCH IF YES
1214
1215                                  ;TEST SETTING OF NON-EXISTENT SECTOR
1216
1217  005322  004767  006162          ERF10:  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1218  005326  005077  006300                  CLR     @RPER
1219  005332  016777  006146  006306          MOV     EXP$,@RPDA      ;LOAD SECTOR ADDR
1220  005340  012777  177777  006272          MOV     #-1,@RPWC
1221  005346  012777  000003  006260          MOV     #3,@RPCS        ;WRITE
1222  005354  012767  060001  006242          MOV     #B14+B13+B0,TPL
1223  005362  004767  006174                  JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
1224  005366  032777  004000  006236          BIT     #B11,@RPER      ;DID NXS SET?
1225  005374  001012                          BNE     1$
1226  005376  104400                          HLT                     ;NXS DID NOT SET WITH ILLEGAL ADDR
1227  005400  004567  173660                  JSR     R5,PRINT$       ;PRINT MESSAGE
1228  005404  013757                          MES10
1229  005406  016767  006072  174146          MOV     EXP$,TTY
1230  005414  004767  173724                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
1231  005420  000413                          BR      2$
1232  005422  032777  040000  006204  1$:     BIT     #B14,@RPCS      ;DID HARD ERROR SET AFTER NXS?
1233  005430  001002                          BNE     3$
1234  005432  104400                          HLT                     ;HARD ERROR DID NOT SET AFTER NXS
1235  005434  000405                          BR      2$
1236  005436  032777  100000  006170  3$:     BIT     #B15,@RPCS      ;DID ERROR SET AFTER NXS?
1237  005444  001001                          BNE     2$
1238  005446  104400                          HLT                     ;ERROR DID NOT SET AFTER NXS
1239  005450  012767  005322  173324  2$:     MOV     #ERF10,LAD
1240  005456  104000                          SCOPE
1241  005460  005267  006020                  INC     EXP$            ;UPDATE ADDRESS
1242  005464  022767  000020  006012          CMP     #20,EXP$        ;IS PATTERN EXHAUSTED?
1243  005472  001313                          BNE     ERF10           ;BRANCH IF NO
1244
1245                                  ;TEST THE SETTING OF PROGRAM ERROR
1246
1247  005474  005077  006132          ERF11:  CLR     @RPER
1248  005500  004767  006004                  JSR     PC,CLRP         ;CLEAR RP11
1249  005504  012777  000003  006122          MOV     #3,@RPCS        ;ISSUE WRITE
1250  005512  012767  060001  006104          MOV     #B14+B13+B0,TPL
1251  005520  004767  006036                  JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
1252  005524  032777  002000  006100          BIT     #B10,@RPER      ;DID PROGRAM ERROR SET?
1253  005532  001002                          BNE     1$
1254  005534  104400                          HLT                     ;PROGRAM ERROR DID NOT SET
1255  005536  000413                          BR      2$              ;WITH WORD COUNT 0
1256  005540  032777  040000  006066  1$:     BIT     #B14,@RPCS      ;IS HARD ERRC? SET?
1257  005546  001002                          BNE     3$
```

```
1258  005550  104400                          HLT                     ;HARD ERROR DID NOT SET
1259  005552  000405                          BR      2$
1260  005554  032777  100000  006052  3$:     BIT     #B15,@RPCS      ;DID ERROR SET?
1261  005562  001001                          BNE     2$
1262  005564  104400                          HLT                     ;ERROR DID NOT SET
1263
1264  005566  012767  005474  173206  2$:     MOV     #ERF11,LAD
1265  005574  104000                          SCOPE
1266
1267                                  ;TEST SETTING OF MODE ERROR
1268
1269  005576  004767  005706          ERF12:  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1270  005602  005077  006024                  CLR     @RPER
1271  005606  012777  177777  006024          MOV     #-1,@RPWC
1272  005614  012777  004003  006012          MOV     #4003,@RPCS     ;WRITE HEADER IN PDP11 MODE
1273  005622  012767  060001  005774          MOV     #B14+B13+B0,TPL
1274  005630  004767  005726                  JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
1275  005634  032777  000400  005770          BIT     #B8,@RPER       ;DID MODE ERROR SET?
1276  005642  001002                          BNE     1$
1277  005644  104400                          HLT                     ;MODE ERROR DID NOT SET
1278  005646  000413                          BR      2$
1279  005650  032777  040000  005756  1$:     BIT     #B14,@RPCS      ;DID HARD ERROR SET?
1280  005656  001002                          BNE     3$
1281  005660  104400                          HLT                     ;HARD ERROR DID NOT SET
1282  005662  000405                          BR      2$
1283  005664  032777  100000  005742  3$:     BIT     #B15,@RPCS      ;DID ERROR SET
1284  005672  001001                          BNE     2$
1285  005674  104400                          HLT                     ;ERROR DID NOT SET
1286  005676  012767  005576  173076  2$:     MOV     #ERF12,LAD
1287  005704  104000                          SCOPE
1288
```

```
1299
1290                                    .SBTTL  ***RPCS FUNCTION TEST***
1291
1292                                    ;CHECK ABILITY OF THE ATTENTION FLAGS TO INTERRUPT
1293
1294  005706  012767  000001  005570  CSF1:   MOV     #1,EXP$         ;STARTING TEST PATTERN
1295  005714  005077  005740          2$:     CLR     @RPM3
1296  005720  012737  000340  000256          MOV     #PRI7,@#STATUS
1297  005726  012737  006020  000254          MOV     #1$,@#VECTOR
1298  005734  004767  005550                  JSR     PC,CLRP         ;CLEAR RP11
1299  005740  005037  177776                  CLR     @#PSW           ;CLEAR PRIORITY LEVEL
1300  005744  052777  020000  005662          BIS     #B13,@RPCS      ;ENABLE ATTENTION INTERRUPT
1301  005752  016777  005526  005674          MOV     EXP$,@RPM2      ;SET ATTENTION BIT AND
1302  005760  000240                          NOP                     ;WAIT FOR INTERRUPT
1303  005762  000240                          NOP
1304  005764  000240                          NOP
1305  005766  012777  000340  172002          MOV     #PRI7,@PSW      ;LOCKOUT INTERRUPTS
1306  005774  104400                          HLT                     ;NO INTERRUPT FROM ATTENTION BIT
1307  005776  004567  173262                  JSR     R5,PRINT$       ;PRINT MESSAGE
1308  006002  013757                          MES10
1309  006004  016767  005474  173550          MOV     EXP$,TTY
1310  006012  004767  173326                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
1311  006016  000412                          BR      3$
1312  006020  012706  000500          1$:     MOV     #STKPTR,SP      ;RESTORE STACK
1313  006024  016777  005454  005576          MOV     EXP$,@RPDS      ;CLEAR ATTENTION BIT
1314  006032  032777  020000  005574          BIT     #B13,@RPCS      ;DID AEI CLEAR?
1315  006040  001401                          BEQ     3$
1316  006042  104400                          HLT                     ;ATTENTION INTERRUPT ENABLE DID NOT
1317                                                                  ;CLEAR WHEN ATTENTION BIT WAS CLEARED
1318  006044  012767  005714  172730  3$:     MOV     #2$,LAD
1319  006052  104000                          SCOPE
1320  006054  006167  005424                  ROL     EXP$            ;SHIFT TEST PATTERN
1321  006060  032767  000400  005416          BIT     #B8,EXP$        ;PATTERN EXCEEDED?
1322  006066  001712                          BEQ     2$              ;BRANCH IF NO
1323
1324
1325                                    ;TEST FOR TWO ATTENTION INTERRUPTS
1326
1327  006070  012737  000340  000256  TSTAT:  MOV     #PRI7,@#STATUS
1328  006076  012737  006150  000254          MOV     #1$,@#VECTOR
1329  006104  004767  005400                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1330  006110  005037  177776                  CLR     @#PSW           ;LOWER PROCESSOR PRIORITY
1331  006114  052777  020000  005512          BIS     #B13,@RPCS      ;ENABLE ATTENTION INTERRUPT
1332  006122  012777  000003  005524          MOV     #3,@RPM2        ;SET ATTENTION BITS
1333  006130  000240                          NOP
1334  006132  000240                          NOP
1335  006134  000240                          NOP
1336  006136  012737  000340  177776          MOV     #PRI7,@#PSW     ;RAISE PROCESSOR PRIORITY
1337  006144  104400                          HLT                     ;RP11C DID NOT INTERRUPT WITH ATTENTION
1338                                                                  ;BITS 0 AND 1 SET
1339  006146  000424                          BR      2$
1340  006150  012706  000500          1$:     MOV     #STKPTR,SP      ;RESTORE STACK
1341  006154  012737  006220  000254          MOV     #2^,@#VECTOR
1342  006162  005037  177776                  CLR     @#PSW
1343  006166  052777  020000  005440          BIS     #B13,@RPCS      ;ENABLE ATTENTION INTERRUPT
1344  006174  012777  000001  005426          MOV     #1,@RPDS        ;CLEAR ATTENTION BIT ZERO
```

```
1345  006202  000240                        NOP
1346  006204  000240                        NOP
1347  006206  000240                        NOP
1348  006210  012737  000340  177776        MOV     #PRI7,@#PSW     ;RAISE PROCESSOR PRIORITY
1349  006216  104400                        HLT                     ;ATTENTION BIT 1 DID NOT INTERRUPT
1350  006220  012706  000500          2$:   MOV     #STKPTR,SP      ;RESTORE THE STACK
1351  006224  012767  006070  172550        MOV     #TSTAT,LAD
1352  006232  104000                        SCOPE
1353
1354
1355                                  ;TEST INTERRUPT ON READY
1356
1357  006234  000005                  CSF2: RESET
1358  006236  012737  006276  000254        MOV     #2$,@#VECTOR    ;RETURN VECTOR
1359  006244  012737  000040  177776        MOV     #PRI1,@#PSW     ;ALLOW INTERRUPTS
1360  006252  052777  000100  005354        BIS     #B6,@RPCS       ;ENABLE INTERRUPT ON READY
1361  006260  000240                        NOP
1362  006262  000240                        NOP
1363  006264  000240                        NOP
1364  006266  012737  000340  177776        MOV     #PRI7,@#PSW     ;LOCKOUT INTERRUPTS
1365  006274  104400                        HLT                     ;NO READY INTERRUPT
1366  006276  012706  000500          2$:   MOV     #STKPTR,SP
1367  006302  012767  006234  172472        MOV     #CSF2,LAD
1368  006310  104000                        SCOPE
1369
1370                                  ;CHECK THAT INTERRUPTS DON'T OCCUR WITHOUT INTERRUPT ENABLES
1371
1372  006312  000005                  CSF3: RESET
1373  006314  012737  006352  000254        MOV     #1$,@#VECTOR    ;INTERRUPT VECTOR
1374  006322  112777  000377  005324        MOVB    #377,@RPM2      ;SET ATTENTION BITS
1375  006330  005037  177776                CLR     @#PSW           ;ALLOW INTERRUPTS AND
1376  006334  000240                        NOP                     ;WAIT AWHILE
1377  006336  000240                        NOP
1378  006340  000240                        NOP
1379  006342  012737  000340  177776        MOV     #PRI7,@#PSW     ;LOCKOUT INTERRUPTS
1380  006350  000403                        BR      2$
1381  006352  012706  000500          1$:   MOV     #STKPTR,SP      ;RESTORE THE STACK
1382  006356  104400                        HLT                     ;INTERRUPT RECEIVED WITHOUT
1383  006360  012767  006312  172414  2$:   MOV     #CSF3,LAD       ;ANY CONTROLLER ENABLES
1384  006366  104000                        SCOPE
1385
1386                                  ;TEST INTERRUPT AT PRIORITY LEVEL 2
1387
1388  006370  004767  005114                CSF4: JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1389  006374  012737  006434  000254        MOV     #1$,@#VECTOR    ;SETUP TRAP VECTOR
1390  006402  052777  000100  005224        BIS     #B6,@RPCS       ;ENABLE RF11 INTERRUPT
1391  006410  012737  000100  177776        MOV     #PRI2,@#PSW     ;LOWER PROCESSOR PRIORITY
1392
1393  006416  000240                        NOP
1394  006420  000240                        NOP
1395  006422  000240                        NOP
1396  006424  012737  000340  177776        MOV     #PRI7,@#PSW
1397  006432  104400                        HLT                     ;NO READY INTERRUPT AT PRIORITY 2
1398  006434  012706  000500          1$:   MOV     #STKPTR,SP
1399  006440  012767  006370  172334        MOV     #CSF4,LAD
1400  006446  104000                        SCOPE
```

```
1401
1402                                        ;TEST INTERRUPT AT PRIORITY LEVEL 3
1403
1404  006450  004767  005034          CSF5:   JSR     PC,CLRP         ;CLEAR RF11
1405  006454  012737  006514  000254          MOV     #1$,@#VECTOR    ;SETUP INTERRUPT VECTOR
1406  006462  052777  000100  005144          BIS     #B6,@RPCS       ;ENABLE READY INTERRUPT
1407  006470  012737  000140  177776          MOV     #PRI3,@#PSW     ;LOWER PROCESSOR LEVEL
1408  006476  000240                          NOP
1409  006500  000240                          NOP
1410  006502  000240                          NOP
1411  006504  012737  000340  177776          MOV     #PRI7,@#PSW
1412  006512  104400                          HLT                     ;NO INTERRUPT AT LEVEL 3
1413  006514  012706  000500          1$:     MOV     #STKPTR,SP      ;RESTORE STACK
1414  006520  012767  006450  172254          MOV     #CSF5,LAD
1415  006526  104000                          SCOPE
1416
1417                                        ;TEST INTERRUPT AT PRIORITY LEVEL 4
1418
1419  006530  004767  004754          CSF6:   JSR     PC,CLRP         ;CLEAR RF11
1420  006534  012737  006574  000254          MOV     #1$,@#VECTOR    ;SETUP INTERRUPT VECTOR
1421  006542  052777  000100  005064          BIS     #B6,@RPCS       ;ENABLE READY INTERRUPT
1422  006550  012737  000200  177776          MOV     #PRI4,@#PSW     ;LOWER PROCESSOR LEVEL
1423  006556  000240                          NOP
1424  006560  000240                          NOP
1425  006562  000240                          NOP
1426  006564  012737  000340  177776          MOV     #PRI7,@#PSW
1427  006572  104400                          HLT                     ;NO READY INTERRUPT AT LEVEL 4
1428  006574  012706  000500          1$:     MOV     #STKPTR,SP      ;RESTORE STACK
1429  006600  012767  006530  172174          MOV     #CSF6,LAD
1430  006606  104000                          SCOPE
1431
1432                                        ;TEST THAT PRIORITY LEVEL 5 LOCKS OUT INTERRUPTS
1433
1434  006610  004767  004674          CSF7:   JSR     PC,CLRP         ;CLEAR RF11
1435  006614  012737  006654  000254          MOV     #1$,@#VECTOR    ;SETUP INTERRUPT VECTOR
1436  006622  052777  000100  005004          BIS     #B6,@RPCS       ;ENABLE INTERRUPT ON READY
1437  006630  012737  000240  177776          MOV     #PRI5,@#PSW     ;SET PRIORITY LEVEL TO 5
1438  006636  000240                          NOP
1439  006640  000240                          NOP
1440  006642  000240                          NOP
1441  006644  012737  000340  177776          MOV     #PRI7,@#PSW
1442  006652  000403                          BR      2$
1443  006654  012706  000500          1$:     MOV     #STKPTR,SP      ;RESTORE STACK
1444  006660  104400                          HLT                     ;INTERRUPT RECEIVED AT PRIORITY LEVEL 5
1445  006662  012767  006610  172112  2$:     MOV     #CSF7,LAD
1446  006670  104000                          SCOPE
1447
1448                                        ;TEST THAT PRIORITY LEVEL 6 LOCKS OUT INTERRUPTS
1449
1450  006672  004767  004612          CSF10:  JSR     PC,CLRP         ;CLEAR RF11
1451  006676  012737  006736  000254          MOV     #1$,@#VECTOR    ;TRAP VECTOR
1452  006704  052777  000100  004722          BIS     #B6,@RPCS       ;ENABLE READY INTERRUPT
1453  006712  012737  000300  177776          MOV     #PRI6,@#PSW     ;SET PRIORITY LEVEL TO 6
1454  006720  000240                          NOP
1455  006722  000240                          NOP
1456  006724  000240                          NOP
```

```
1457  006726  012737  000340  177776          MOV     #PRI7,@#PSW
1458  006734  000403                          BR      2$
1459  006736  012706  000500          1$:     MOV     #STKPTR,SP      ;RESTORE STACK
1460  006742  104400                          HLT                     ; INTERRUPT RECEIVED AT LEVEL 6
1461  006744  012767  006672  172030  2$:     MOV     #CSF10,LAD
1462  006752  104000                          SCOPE
1463
1464                                  ;TEST THAT PRIORITY LEVEL 7 LOCKS OUT INTERRUPTS
1465
1466  006754  004767  004530          CSF11:  JSR     PC,CLRP         ;CLEAR RF11
1467  006760  012737  007004  000254          MOV     #1$,@#VECTOR    ;SETUP VECTOR INTERRUPT
1468  006766  052777  000100  004640          BIS     #B6,@RPCS       ;ENABLE READY INTERRUPT
1469  006774  000240                          NOP
1470  006776  000240                          NOP
1471  007000  000240                          NOP
1472  007002  000403                          BR      2$
1473  007004  012706  000500          1$:     MOV     #STKPTR,SP
1474  007010  104400                          HLT                     ;INTERRUPT RECEIVED AT LEVEL 7
1475  007012  012767  006754  171762  2$:     MOV     #CSF11,LAD
1476  007020  104000                          SCOPE
1477
1478                                  ;TEST THE CLEARING AND SETTING OF READY (RPCS-B7)
1479
1480  007022  004767  004462          CSF12:  JSR     PC,CLRP         ;CLEAR RF11
1481  007026  012777  000003  004600          MOV     #3,@RPCS        ;ISSUE WRITE
1482  007034  105777  004574                  TSTB    @RPCS           ;IS READY SET?
1483  007040  100002                          BPL     1$              ;BRANCH IF NO
1484  007042  104400                          HLT                     ;READY DID NOT CLEAR WITH GO H
1485  007044  000411                          BR      2$
1486  007046  012767  060001  004550  1$:     MOV     #B14+B13+B0,TPL
1487  007054  004767  004502                  JSR     PC,T3P          ;GENERATE 3 TIMING PULSES
1488  007060  105777  004550                  TSTB    @RPCS           ;DID READY SET?
1489  007064  100401                          BMI     2$
1490  007066  104400                          HLT                     ;READY DID NOT SET AT END OF OPERATION
1491  007070  012767  007022  171704  2$:     MOV     #CSF12,LAD
1492  007076  104000                          SCOPE
1493
```

```
                                        .SBTTL  ***SUCA FUNCTION TEST***

007100  000005                  CAF1:   RESET
007102  005067  004376                  CLR     EXP$                ;START TEST PATTERN AT 0
007106  004767  004376          CAL1:   JSR     PC,CLRP             ;CLEAR THE CONTROLLER
007112  116777  004366  004536          MOVB    EXP$,@RPM21         ;LOAD MAINT CYLINDER ADDR
007120  017767  004540  004360          MOV     @SUCA,REC$          ;GET DISK CYLINDER ADDR
007126  026767  004352  004352          CMP     EXP$,REC$           ;IS SUCA CORRECT?
007134  001401                          BEQ     1$
007136  104401                          HLT     +1                  ;SUCA INCORRECT

007140  012767  007106  171634  1$:     MOV     #CAL1,LAD
007146  104000                          SCOPE
007150  005267  004330                  INC     EXP$                ;UPDATE CYLINDER ADDR.
007154  032767  000400  004322          BIT     #B8,EXP$            ;IS PATTERN EXCEEDED?
007162  001751                          BEQ     CAL1                ;BRANCH IF NO
```

```
                                        .SBTTL  ***RPDA FUNCTION TEST***

007164  004767  004320        DAF1:   JSR     PC,CLRP         ;CLEAR RF11
00717C  012700  000001                MOV     #1,R0
007174  004767  004372                JSR     PC,INDEXP       ;GENERATE 1 INDEX PULSE
007200  052777  000400  004452        BIS     #B8,@RPM3       ;GENERATE 1 SECTOR PULSE
007206  042777  000400  004444        BIC     #B8,@RPM3
007214  052777  000400  004436        BIS     #B8,@RPM3
007222  042777  000400  004430        BIC     #B8,@RPM3       ;FORCE SOT TO COUNT OF 1
007230  012700  000001                MOV     #1,R0
007234  004767  004332                JSR     PC,INDEXP       ;GENERATE 1 INDEX PULSE
007240  052777  000400  004412        BIS     #B8,@RPM3
007246  042777  000400  004404        BIC     #B8,@RPM3       ;CLEAR SOT
007254  017700  004366                MOV     @RPDA,R0
007260  006000                        ROR     R0
007262  006000                        ROR     R0
007264  006000                        ROR     R0
007266  006000                        ROR     R0
007270  042700  177760                BIC     #-17-1,R0       ;CLEAR UNWANTED BITS
007274  005700                        TST     R0              ;WAS SOT CLEARED?
007276  001401                        BEQ     1$
007300  104400                        HLT                     ;SOT DID NOT CLEAR WITH INDEX PULSE
007302  012767  007164  171472  1$:   MOV     #DAF1,LAD
007310  104000                        SCOPE

                                        ;TEST ABILITY OF SOT TO COUNT

007312  004767  004172        DAF2:   JSR     PC,CLRP         ;CLEAR RP11C
007316  005067  004162                CLR     EXP$            ;EXPECT STARTING ADDR OF 0
007322  012700  000001                MOV     #1,R0
007326  004767  004240                JSR     PC,INDEXP       ;GENERATE ONE INDEX PULSE
007332  052777  000400  004320  2$:   BIS     #B8,@RPM3       ;GENERATE ONE SECTOR PULSE
007340  042777  000400  004312        BIC     #B8,@RPM3
007346  017767  004274  004132        MOV     @RPDA,REC$      ;GET SOT
007354  006067  004126                ROR     REC$
007360  006067  004122                ROR     REC$
007364  006067  004116                ROR     REC$
007370  006067  004112                ROR     REC$
007374  042767  177760  004104        BIC     #-17-1,REC$     ;CLEAR UNWANTED BITS
007402  026767  004076  004076        CMP     EXP$,REC$       ;IS CONTENTS OF SOT CORRECT?
007410  001401                        BEQ     1$
007412  104401                        HLT     +1              ;CONTENTS OF SOT INCORRECT
007414  012767  007312  171360  1$:   MOV     #DAF2,LAD
007422  104000                        SCOPE
007424  005267  004054                INC     EXP$            ;UPDATE TEST ADDR
007430  022767  000012  004046        CMP     #12,EXP$        ;ADDR EXHAUSTED?
007436  001335                        BNE     2$              ;BRANCH IF NO
```

```
                                        .SBTTL  *** SILO MEMORY TEST ***

                                        ;THIS TEST CHECKS THE SILO MEMORY IN MAINTENANCE
                                        ;MODE.  IF INREADY IS SET, DATA IS OUTPUT TO REGISTER
                                        ;776736 WHICH IS THE SILO MEMORY.  AFTER THE DATA FILTERS
                                        ;THRU THE MEMORY, OUTREADY GOES TRUE.  THE DATA IS READ
                                        ;BACK AND COMPARED.

007440  012704  177760          SILO:   MOV     #-20,R4         ;SET UP ITERATION COUNT FOR SILO TEST
007444  004767  004040          SILOR:  JSR     PC,CLRP         ;CLEAR THE RF11C
007450  012767  000001  004026          MOV     #1,EXP$         ;INITIATE FLOATING ONE PATTERN
007456  005000                        CLR     R0              ;PATTERN FLOG
007460  000005                1$:     RESET
007462  032777  004000  004162          BIT     #B11,@RPM1      ;IS INREADY SET?
007470  001002                        BNE     6$              ;BRANCH IF SET
007472  104400                        HLT                     ;SILO INREADY IS NOT SET AFTER CLEAR
007474  000425                        BR      3$
007476  016777  004002  004156  6$:     MOV     EXP$,@RPS0      ;LOAD DATA IN SILO
007504  012701  000060                MOV     #60,R1          ;WAIT FOR OUT READY TO SET
007510  005301                8$:     DEC     R1
007512  001376                        BNE     8$
007514  032777  010000  004130  2$:     BIT     #B12,@RPM1      ;IS OUTREADY SET?
007522  001002                        BNE     7$              ;BRANCH IF SET
007524  104400                        HLT                     ;SILO OUTREADY IS NOT SET
007526  000410                        BR      3$
007530  017767  004126  003750  7$:     MOV     @RPS0,REC$      ;GET DATA BACK FROM SILO
007536  026767  003742  003742          CMP     EXP$,REC$       ;IS DATA CORRECT?
007544  001401                        BEQ     3$
007546  104403                        HLT     +3              ;DATA READ FROM SILO INCORRECT
007550  012767  007460  171224  3$:     MOV     #1$,LAD
007556  104000                        SCOPE
007560  005700                        TST     R0              ;ARE WE FLOATING A ONE?
007562  001013                        BNE     4$              ;BRANCH IF NO
007564  000241                        CLC
007566  006167  003712                ROL     EXP$            ;ROTATE PATTERN
007572  103401                        BCS     5$              ;BRANCH IF PATTERN EXCEEDED
007574  000731                        BR      1$
007576  012700  000001          5$:     MOV     #1,R0           ;SET PATTERN FLAG
007602  012767  077777  003674          MOV     #077777,EXP$    ;FLOATING ZERO PATTERN
007610  000723                        BR      1$
007612  000241                4$:     CLC
007614  006267  003664                ASR     EXP$            ;SHIFT FLOATING ZERO
007620  052767  100000  003656          BIS     #B15,EXP$
007626  103714                        BCS     1$
                                        ;ENSURE THAT THE SILO MEMORY CAN HOLD 64 DISCREET NUMBERS
                                        ;AT ONE TIME.  AFTER LOADING THE 64 NUMBERS SIGNAL INREADY
                                        ;SHOULD CLEAR INDICATING THE SILO IS FULL.  THE SILO IS THEN
                                        ;READ OUT EXPECTING SEQUENTIAL NUMBERS OF 1 THRU 100 OCTAL. AT
                                        ;THIS TIME OUT READY SHOULD CLEAR.

007630  004767  003654          SILO1:  JSR     PC,CLRP         ;CLEAR THE RP11C
007634  012767  000001  003642          MOV     #1,EXP$         ;INITIAL TEST PATTERN
007642  032777  004000  004002  1$:     BIT     #B11,@RPM1      ;IS INREADY SET?
007650  001002                        BNE     2$              ;BRANCH IF SET
007652  104400                        HLT                     ;SILO INREADY SHOULD BE SET
007654  000454                        BR      3$
```

```
1615  007656  016777  003622  003776  2$:     MOV     EXP$,@RPS0      ;LOAD PATTERN IN SILO
1616  007664  005267  003614                  INC     EXP$            ;UPDATE PATTERN
1617  007670  022767  000101  003606          CMP     #101,EXP$       ;IS THE SILO FULL?
1618  007676  001361                          BNE     1$              ;BRANCH NOT FULL
1619  007700  032777  004000  003744          BIT     #B11,@RPM1      ;DID INREADY CLEAR?
1620  007706  001402                          BEQ     7$              ;BRANCH IF YES
1621  007710  104400                          HLT                     ;SILO INREADY DID NOT CLEAR
1622  007712  000435                          BR      3$              ;AFTER SILO WAS FILLED
1623  007714  012767  000001  003562  7$:     MOV     #1,EXP$         ;RESET PATTERN
1624  007722  032777  010000  003722  6$:     BIT     #B12,@RPM1      ;IS OUTREADY SET?
1625  007730  001002                          BNE     4$              ;BRANCH IF SET
1626  007732  104400                          HLT                     ;SILO OUTREADY SHOULD BE SET
1627  007734  000424                          BR      3$
1628  007736  017767  003720  003542  4$:     MOV     @RPS0,REC$      ;READ SILO MEMORY
1629  007744  026767  003534  003534          CMP     EXP$,REC$       ;IS DATA CORRECT?
1630  007752  001402                          BEQ     5$              ;BRANCH IS EQUAL
1631  007754  104403                          HLT     +3              ;INCORRECT DATA RECEIVED FROM SILO
1632  007756  000413                          BR      3$
1633  007760  005267  003520          5$:     INC     EXP$            ;UPDATE EXPECTED PATTERN
1634  007764  022767  000101  003512          CMP     #101,EXP$       ;HAVE 6+ WORDS BEEN READ
1635  007772  001353                          BNE     6$              ;BRANCH IF NO
1636  007774  032777  010000  003650          BIT     #B12,@RPM1      ;HAS SILO OUTREADY CLEARED
1637  010002  001401                          BEQ     3$              ;BRANCH IF CLEAR
1638  010004  104400                          HLT                     ;SILO SHOULD BE EMPTY BUT
1639                                                                  ;OUTREADY IS STILL SET
1640  010006  012767  007630  170766  3$:     MOV     #SILO1,LAD
1641  010014  104000                          SCOPE
1642  010016  005204                          INC     R4              ;INCREMENT ITERATION COUNT
1643  010020  001402                          BEQ     SEEK
1644  010022  000167  177416                  JMP     SILOR           ;LOOP 20 TIMES
1645
1646                                  ;TEST THE OPERATION OF A SEEK COMMAND IN MAINTENANCE MODE.
1647                                  ;ISSUE A SEEK COMMAND AND CHECK THE SETTING OF SET CYLINDER
1648                                  ;RESET HEAD, SET HEAD,SEEK START AND CAR BITS 0 THRU 7.
1649                                  ;ALL THESE SIGNALS ARE FOUND ON BUS OUT CONTROL.
1650  010026  000005                  SEEK:   RESET
1651  010030  005004                          CLR     R4
1652  010032  017767  003614  003446  10$:    MOV     @RPM1,REC$      ;GET CONTENTS OF RPM1
1653  010040  042767  174000  003440          BIC     #174000,REC$    ;CLEAR UNWANTED BITS
1654  010046  001404                          BEQ     1$              ;BRANCH IF RESULT IS ZERO
1655  010050  005067  003430                  CLR     EXP$
1656  010054  104403                          HLT     +3              ;SOME BUS OUT SIGNALS TO THE
1657  010056  000553                          BR      2$              ;RP03 ARE SET AFTER RESET
1658  010060  012777  000376  003556  1$:     MOV     #376,@RPCA      ;LOAD CYCL 376 INTO RPCA
1659  010066  005704                          TST     R4
1660  010070  001403                          BEQ     4$
1661  010072  012777  000001  003544          MOV     #1,@RPCA
1662  010100  112777  000017  003542  4$:     MOVB    #17,@RPDA1      ;LOAD TRACK ADDR
1663  010106  005704                          TST     R4              ;IS THIS FIRST PASS
1664  010110  001403                          BEQ     3$              ;BRANCH IF YES
1665  010112  112777  000020  003530          MOVB    #20,@RPDA1      ;SET HIGH ORDER BIT OF TRACK ADDR
1666  010120  012777  000011  003506  3$:     MOV     #11,@RPCS       ;ISSUE SEEK COMMAND
1667  010126  012767  060001  003470          MOV     #B14+B13+B0.TPL
1668  010134  012700  000005                  MOV     #5,R0
1669  010140  004767  003400                  JSR     PC,TIMEP        ;GENERATE 5 TIMING PULSES
1670  010144  017767  003502  003334          MOV     @RPM1,REC$      ;GET CONTROL LINER FROM RPM1
```

```
1671  010152  042767  174000  003326          BIC     #174000,REC$    ;CLEAR UNWANTED BITS
1672  010160  012767  000577  003316          MOV     #577,EXP$       ;LOAD EXPECTED VALUE OF RPM1
1673  010166  005704                          TST     R4              ;IS THIS FIRST PASS
1674  010170  001403                          BEQ     5$              ;BRANCH IF YES
1675  010172  012767  000600  003304          MOV     #600,EXP$
1676  010200  026767  003300  003300  5$:     CMP     EXP$,REC$       ;WERE CONTENTS OF RPM1 CORRECT?
1677  010206  001402                          BEQ     6$              ;BRANCH IF YES
1678  010210  104403                          HLT     +3              ;THE CONTROL SIGNAL SET CYLINDER
1679  010212  000475                          BR      2$              ;AND THE CONTENTS OF CAR SHOULD
1680                                                                  ;BE ON THE BUS OUT LINES
1681  010214  012700  000004          6$:     MOV     #4,R0
1682  010220  004767  003320                  JSR     PC,TIMEP        ;GENERATE 4 CLOCK PULSES
1683  010224  017767  003422  003254          MOV     @RPM1,REC$      ;GET CONTENTS OF RPM1
1684  010232  042767  174000  003246          BIC     #174000,REC$    ;CLEAR UNWANTED BITS
1685  010240  012767  002010  003236          MOV     #B10+B3,EXP$    ;LOAD EXPECTED VALUE OF RPM1
1686  010246  026767  003232  003232          CMP     EXP$,REC$       ;ARE CONTENTS OF RPM1 CORRECT?
1687  010254  001402                          BEQ     7$              ;BRANCH IF YES
1688  010256  104403                          HLT     +3              ;THE CONTROL SIGNAL RESET HEAD
1689  010260  000452                          BR      2$              ;SHOULD BE ON THE BUS OUT LINES
1690  010262  012700  000004          7$:     MOV     #4,R0
1691  010266  004767  003252                  JSR     PC,TIMEP        ;GENERATE 4 CLOCK PULSES
1692  010272  017767  003354  003206          MOV     @RPM1,REC$      ;GET CONTENTS OF RPM1
1693  010300  042767  174000  003200          BIC     #174000,REC$    ;CLEAR UNWANTED
1694  010306  012767  001360  003170          MOV     #1360,EXP$      ;LOAD EXPECTED VALUE OF RPM1
1695  010314  005704                          TST     R4              ;IS THIS THE FIRST PASS
1696  010316  001403                          BEQ     8$              ;BRANCH IF YES
1697  010320  012767  001010  003156          MOV     #1010,EXP$
1698  010326  026767  003152  003152  8$:     CMP     EXP$,REC$       ;ARE CONTENTS OF RPM1 CORRECT?
1699  010334  001402                          BEQ     9$
1700  010336  104403                          HLT     +3              ;THE CONTROL SIGNAL SET HEAD AND
1701  010340  000422                          BR      2$              ;THE HEAD ADDRESS SHOULD BE ON
1702                                                                  ;THE BUS OUT LINES
1703  010342  012700  000004          9$:     MOV     #4,R0
1704  010346  004767  003172                  JSR     PC,TIMEP        ;GENERATE 4 CLOCK PULSES
1705  010352  017767  003274  003126          MOV     @RPM1,REC$      ;GET CONTENTS OF RPM1
1706  010360  042767  174000  003120          BIC     #174000,REC$    ;CLEAR UNWANTED BITS
1707  010366  012767  002004  003110          MOV     #B10+B2,EXP$    ;LOAD EXPECTED BITS
1708  010374  026767  003104  003104          CMP     EXP$,REC$       ;ARE CONTENTS OF RPM1 CORRECT?
1709  010402  001401                          BEQ     2$              ;BRANCH IS YES
1710  010404  104403                          HLT     +3              ;THE CONTROL SIGNAL SEEK START
1711                                                                  ;SHOULD BE ON THE BUS OUT LINES
1712  010406  000005          2$:     RESET
1713  010410  012767  010032  170364          MOV     #10$,LAD
1714  010416  104000                          SCOPE
1715  010420  005704                          TST     R4              ;IS THIS FIRST PASS?
1716  010422  001003                          BNE     SEEK1           ;BRANCH IF NO .
1717  010424  005204                          INC     R4
1718  010426  000167  177400                  JMP     10$             ;MAKE SECOND PASS
1719  010432  000240          SEEK1:  NOP
1720
1721                                  ;ISSUE A RESTORE COMMAND AND CHECK THE GENERATION OF
1722                                  ;THE SIGNAL RESTORE ON THE BUS OUT CONTROL LOGIC
1723
1724  010434  000005          RESTOR: RESET
1725  010436  017767  003210  003042  3$:     MOV     @RPM1,REC$      ;GET CONTENTS OF RPM1
1726  010444  042767  174000  003034          BIC     #174000,REC$    ;CLEAR UNWANTED BITS
```

```
1727  010452  001404                          BEQ     1$              ;BRANCH IF RESULT IS YES
1728  010454  005067  003024                  CLR     EXP$
1729  010460  104403                          HLT     +3              ;SOME BUS OUT SIGNALS TO THE
1730  010462  000425                          BR      2$              ;RP03 ARE SET AFTER RESET
1731  010464  012777  000015  003142  1$:     MOV     #15,@RPCS       ;ISSUE HOME COMMAND
1732  010472  012700  000004                  MOV     #4,R0
1733  010476  004767  003042                  JSR     PC,TIMEP        ;GENERATE 4 TIMING PULSES
1734  010502  017767  003144  002776          MOV     @RPM1,REC$      ;GET CONTROL LINES FROM RPM1
1735  010510  042767  174000  002770          BIC     #174000,REC$    ;CLEAR UNWANTED BITS
1736  010516  012767  002100  002760          MOV     #B6+B10,EXP$    ;LOAD EXPECTED VALUE OF RPM1
1737  010524  026767  002754  002754          CMP     EXP$,REC$       ;WERE CONTENTS OF RPM1 CORRECT?
1738  010532  001401                          BEQ     2$              ;BRANCH IF YES
1739  010534  104403                          HLT     +3              ;ISSUED HOME COMMAND-EXPECTED
1740                                                                  ;RESTORE AND CONTROL TO BE SET
1741                                                                  ;ON BUS OUT LINES
1742  010536  000005                  2$:     RESET
1743  010540  012767  010436  170234          MOV     #3$,LAD
1744  010546  104000                          SCOPE
1745
1746                                  ;ISSUE A READ COMMAND WITH NO SEEK IMPLIED AND CHECK THE GENERATION
1747                                  ;OF THE SIGNAL READ IN THE BUS OUT LOGIC
1748
1749  010550  000005                  READT:  RESET
1750  010552  012777  177777  003060  2$:     MOV     #-1,@RPWC       ;LOAD WORD COUNT
1751  010560  012777  000017  003046          MOV     #17,@RPCS       ;ISSUE READ COMMAND
1752  010566  012700  000004                  MOV     #4,R0
1753  010572  004767  002746                  JSR     PC,TIMEP
1754  010576  012767  060401  003020          MOV     #B14+B13+B8+B0,TPL
1755  010604  012700  000003                  MOV     #3,R0
1756  010610  004767  002730                  JSR     PC,TIMEP        ;GENERATE TIMING PULSE
1757  010614  012767  060001  003002          MOV     #B14+B13+B0,TPL
1758  010622  017767  003024  002656          MOV     @RPM1,REC$      ;GET CONTENTS OF RPM1
1759  010630  042767  174000  002650          BIC     #174000,REC$    ;CLEAR UNWANTED BITS
1760  010636  012767  002042  002640          MOV     #B10+B5+B1,EXP$ ;LOAD EXPECTED VALUE OF RPM1
1761  010644  026767  002634  002634          CMP     EXP$,REC$       ;ARE CONTENTS OF RPM1 CORRECT
1762  010652  001401                          BEQ     1$              ;BRANCH IF YES
1763  010654  104403                          HLT     +3              ;ISSUED READ WITH NO SEEK COMMAND
1764                                                                  ;EXPECTED CONTROL, SELECT HEAD, AND READ
1765                                                                  ;SIGNALS ON THE BUS OUT LINES
1766  010656  000005                  1$:     RESET
1767  010660  012767  010552  170114          MOV     #2$,LAD
1768  010666  104000                          SCOPE
1769
1770                                  ;ISSUE A WRITE FORMAT COMMAND AND CHECK THE GENERATION OF THE
1771                                  ;WRITE SIGNAL IN THE BUS OUT LOGIC
1772
1773  010670  012777  177777  002742  WRITE:  MOV     #-1,@RPWC       ;SET UP WORD COUNT
1774  010676  012777  000001  002742          MOV     #1,@RPDA        ;SELECT SECTOR ONE
1775  010704  012777  014003  002722          MOV     #14003,@RPCS    ;ISSUE WRITE FORMAT COMMAND
1776  010712  012700  000023                  MOV     #23,R0
1777  010716  004767  002622                  JSR     PC,TIMEP        ;GENERATE 19 TIMING PULSES
1778  010722  012700  000001                  MOV     #1,R0
1779  010726  004767  002640                  JSR     PC,INDEXP       ;GENERATE INDEX PULSE
1780  010732  052777  000400  002720          BIS     #B8,@RPM3       ;GENERATE SECTOR PULSE
1781  010740  042777  000400  002712          BIC     #B8,@RPM3
1782  010746  052777  000400  002704          BIS     #B8,@RPM3       ;GENERATE SECTOR PULSE
```

```
 1783  010754  012700  000003                    MOV     #3,R0
 1784  010760  004767  002560                    JSR     PC,TIMEP        ;GENERATE 3 TIMING PULSES
 1785  010764  017767  002662  002514            MOV     @RPM1,REC$      ;GET CONTENTS OF RPM1
 1786  010772  042767  174000  002506            BIC     #174000,REC$    ;CLEAR UNWANTED BITS
 1787  011000  012767  002061  002476            MOV     #B10+B5+B4+B0,EXP$      ;LOAD EXPECTED VALUE OF RPM1
 1788  011006  026767  002472  002472            CMP     EXP$,REC$       ;ARE CONTENTS OF RPM1 CORRECT
 1789  011014  001401                            BEQ     1$              ;BRANCH IF YES
 1790  011016  104403                            HLT     +3              ;ISSUED A WRITE FORMAT COMMAND
 1791                                                                    ;AND EXPECTED CONTROL, SELECT HEADS, ERASE,
 1792                                                                    ;AND WRITE SIGNALS ON THE BUS OUT LOGIC
 1793  011020  000005                    1$:     RESET
 1794  011022  012767  010670  167752            MOV     #WRITE,LAD
 1795  011030  104000                            SCOPE
 1796  011032  005267  167742                    INC     ICNT            ;INCREMENT ITERATION COUNT
 1797  011036  022767  000100  167734            CMP     #100,ICNT       ;64 PASSES YET?
 1798  011044  001402                            BEQ     2$              ;BRANCH IF YES .
 1799  011046  000167  171304                    JMP     ERBT
 1800  011052  005267  002540            2$:     INC     PASSCT          ;INCREMENT PASS COUNT
 1801  011056  004567  170202                    JSR     R5,PRINT$       ;PRINT MESSAGE
 1802  011062  013666                            MES1
 1803  011064  016767  002526  170470            MOV     PASSCT,TTY
 1804  011072  004767  170260                    JSR     PC,PRINTS       ;TYPE LOCATION-SUPRESS ZEROS
 1805  011076  000167  171244                    JMP     LOOP
```

```
                                        .SBTTL  ***SWITCH TEST***

                                        ;THIS TEST IS CALLED WHEN IT IS DESIRED TO CHECK THE RP03 AND
                                        ;THE RP11C SWITCHES.  INITIALLY BEFORE STARTING THE TEST, THE
                                        ;ENABLE-DISABLE SWITCH SHOULD BE ENABLED.  THE DRIVE SHOULD HAVE
                                        ;READ-WRITE ENABLED.  THE WRITE LOCKOUT SWITCH SHOULD BE CLEARED
                                        ;AND FORMATTING AND MAINTENANCE DISABLED.  TO START THE TEST
                                        ;LOAD 250 INTO PC AND START.

011102  012706  000500          SWTST:  MOV     #STKPTR,SP      ;SETUP STACK
011106  004767  002142                  JSR     PC,INIT         ;INITIALIZE VECTORS
011112                          1$:
011112  004567  170146                  JSR     R5,PRINT$       ;PRINT MESSAGE
011116  012112                          MES20                   ;GET NUMBER OF DRIVES
011120  004767  170560                  JSR     PC,READ$        ;INPUT MESSAGE
011124  004767  170732                  JSR     PC,PACK$        ;CONVERT INPUT TO A NUMBER
011130  005767  171200                  TST     NUM$            ;IS RESPONSE 0?
011134  001766                          BEQ     1$              ;BRANCH IF YES
011136  162767  000001  171170          SUB     #1,NUM$
011144  022767  000010  171162          CMP     #10,NUM$        ;IS NUMBER TOO HIGH?
011152  101757                          BLOS    1$              ;BRANCH IF YES
011154  016767  171154  002440          MOV     NUM$,HUNIT      ;SAVE HIGHEST UNIT NO.
011162  005067  002432                  CLR     UNIT
011166  004567  170072                  JSR     R5,PRINT$       ;PRINT MESSAGE
011172  012170                          MES22
011174                          5$:
011174  004567  170064                  JSR     R5,PRINT$       ;PRINT MESSAGE
011200  012147                          MES21                   ;HAVE DRIVE DISABLED
011202  016767  002412  170352          MOV     UNIT,TTY
011210  004767  170142                  JSR     PC,PRINTS       ;TYPE LOCATION-SUPRESS ZEROS
011214  004767  170464                  JSR     PC,READ$        ;INPUT MESSAGE
011220  012777  000001  002406  2$:     MOV     #1,@RPCS        ;CLEAR THE CONTROLLER
011226  116777  002366  002402          MOVB    UNIT,@RPCS1     ;SELECT UNIT
011234  032777  040000  002366          BIT     #B14,@RPDS      ;IS THE UNIT ON LINE?
011242  001402                          BEQ     4$              ;BRANCH IF NO
011244  104400                          HLT                     ;SELECTED UNIT ON LINE WITH THE
011246  000425                          BR      3$              ;DRIVE DISABLED
011250                          4$:
011250  004567  170010                  JSR     R5,PRINT$       ;PRINT MESSAGE
011254  012244                          MES23                   ;HAVE DRIVE ENABLED
011256  016767  002336  170276          MOV     UNIT,TTY
011264  004767  170066                  JSR     PC,PRINTS       ;TYPE LOCATION-SUPRESS ZEROS
011270  004767  170410                  JSR     PC,READ$        ;INPUT MESSAGE
011274  012777  000001  002332          MOV     #1,@RPCS        ;CLEAR THE CONTROLLER
011302  116777  002312  002326          MOVB    UNIT,@RPCS1     ;RESELECT THE UNIT
011310  032777  040000  002312          BIT     #B14,@RPDS      ;IS THE UNIT ON LINE?
011316  001001                          BNE     3$
011320  104400                          HLT                     ;SELECTED UNIT NOT ON LINE WITH
                                        ;THE DRIVE ENABLED
011322  012767  011220  167452  3$:     MOV     #2$,LAD
011330  104000                          SCOPE
011332  026767  002262  002262          CMP     UNIT,HUNIT      ;HAVE ALL UNITS BEEN TESTED?
011340  001403                          BEQ     RDSW            ;BRANCH IF YES
011342  005267  002252                  INC     UNIT            ;UPDATE UNIT
011346  000712                          BR      5$
011350  000005                  RDSW:   RESET
```

```
 1862  011352  005067  002242              CLR     UNIT
 1863  011356                        5$:
 1864  011356  004567  167702              JSR     R5,PRINT$       ;PRINT MESSAGE
 1865  011362  012264                      MES24                   ;HAVE READ ONLY SET
 1866  011364  016767  002230  170170      MOV     UNIT,TTY
 1867  011372  004767  167760              JSR     PC,PRINTS       ;TYPE LOCATION-SUPRESS ZEROS
 1868  011376  004767  170302              JSR     PC,READ$        ;INPUT MESSAGE
 1869  011402  012777  000001  002224  3$: MOV     #1,@RPCS        ;CLEAR THE CONTROLLER
 1870  011410  116777  002204  002220      MOVB    UNIT,@RPCS1     ;SELECT THE UNIT
 1871  011416  032777  000400  002204      BIT     #BB,@RPDS       ;IS SELECTED UNIT WRITE PROTECTED
 1872  011424  001002                      BNE     1$
 1873  011426  104400                      HLT                     ;SELECTED UNIT WRITE PROTECT DID
 1874  011430  000425                      BR      2$              ;NOT SET WITH READ ONLY SET
 1875  011432                        1$:
 1876  011432  004567  167626              JSR     R5,PRINT$       ;PRINT MESSAGE
 1877  011436  012316                      MES25                   ;CLEAR READ ONLY
 1878  011440  016767  002154  170114      MOV     UNIT,TTY
 1879  011446  004767  167704              JSR     PC,PRINTS       ;TYPE LOCATION-SUPRESS ZEROS
 1880  011452  004767  170226              JSR     PC,READ$        ;INPUT MESSAGE
 1881  011456  012777  000001  002150      MOV     #1,@RPCS        ;CLEAR THE CONTROLLER
 1882  011464  116777  002130  002144      MOVB    UNIT,@RPCS1     ;RESELECT THE UNIT
 1883  011472  032777  000400  002130      BIT     #BB,@RPDS       ;IS SELECTED UNIT WRITE PROTECTED
 1884  011500  001401                      BEQ     2$
 1885  011502  104400                      HLT                     ;SELECTED UNIT WRITE PROTECT SET
 1886                                                              ;WITH DRIVE WRITE ENABLED
 1887  011504  012767  011402  167270  2$: MOV     #3$,LAD
 1888  011512  104000                      SCOPE
 1889  011514  026767  002100  002100      CMP     UNIT,HUNIT      ;HAVE ALL DRIVES BEEN TESTED?
 1890  011522  001403                      BEQ     4$              ;BRANCH IF YES
 1891  011524  005267  002070              INC     UNIT            ;UPDATE UNIT
 1892  011530  000712                      BR      5$
 1893  011532  005067  002062        4$:   CLR     UNIT
 1894
 1895                                    ;TEST THE SETTING OF WRITE LOCKOUT AND THE BOUNDARY SWITCHES ON THE
 1896                                    ;RP11C CONTROL PANEL.
 1897
 1898  011536  000005                WRSW: RESET
 1899  011540  012700  000002              MOV     #2,R0           ;INITIATE SWITCH PATTERN
 1900  011544  004567  167514              JSR     R5,PRINT$       ;PRINT MESSAGE
 1901  011550  012352                      MES26                   ;HAVE WRITE LOCKOUT SET
 1902  011552  004567  167536              JSR     R5,PRINT$       ;PRINT MESSAGE
 1903  011556  012425                      MES27
 1904  011560  004767  170120              JSR     PC,READ$        ;INPUT MESSAGE
 1905  011564  005077  002054        3$:   CLR     @RPCA
 1906  011570  032777  000400  002032      BIT     #BB,@RPDS       ;IS SELECTED UNIT WRITE PROTECT SET?
 1907  011576  001002                      BNE     1$
 1908  011600  104400                      HLT                     ;SELECTED UNIT WRITE PROTECT SHOULD
 1909  011602  000410                      BR      2$              ;BE SET-RPCA EQUAL TO ZERO
 1910  011604  012777  000002  002032  1$: MOV     #2,@RPCA        ;PUT A TWO IN RPCA
 1911  011612  032777  000400  002010      BIT     #BB,@RPDS       ;IS SELECTED UNIT WRITE PROTECTED?
 1912  011620  001401                      BEQ     2$
 1913  011622  104400                      HLT                     ;SELECTED UNIT WRITE PROTECT IS SET
 1914                                                              ;WITH RPCA EQUAL TO TWO.
 1915  011624  012767  011564  167150  2$: MOV     #3$,LAD
 1916  011632  104000                      SCOPE
 1917
```

```
1918  011634                          WRSW1:
1919  011634  004567  167424                JSR     R5,PRINT$       ;PRINT MESSAGE
1920  011640  012472                        MES28                   ;SET CYLINDER SWITCHES
1921  011642                          4$:
1922  011642  004567  167416                JSR     R5,PRINT$       ;PRINT MESSAGE
1923  011646  012571                        MES29
1924  011650  000241                        CLC
1925  011652  006000                        ROR     R0
1926  011654  010067  167702                MOV     R0,TTY
1927  011660  004767  167472                JSR     PC,PRINTS       ;TYPE LOCATION-SUPRESS ZEROS
1928  011664  000241                        CLC
1929  011666  006100                        ROL     R0
1930  011670  004767  170010                JSR     PC,READ$        ;INPUT MESSAGE
1931  011674  010077  001744          3$:   MOV     R0,@RPCA        ;LOAD PATTERN INTO RPCA
1932  011700  032777  000400  001722        BIT     #BB,@RPDS       ;IS UNIT WRITE PROTECT SET?
1933  011706  001002                        BNE     1$
1934  011710  104400                        HLT                     ;UNIT WRITE PROTECT IS NOT SET
1935  011712  000410                        BR      2$              ;RPCA EQUALS CONTENTS OF BOUNDARY
1936                                                                ;SWITCHES
1937  011714  062777  000002  001722  1$:   ADD     #2,@RPCA        ;INCREMENT CYLINDER ADDR
1938  011722  032777  000400  001700        BIT     #BB,@RPDS       ;IS UNIT WRITE PROTECT SET?
1939  011730  001401                        BEQ     2$
1940  011732  104400                        HLT                     ;UNIT WRITE PROTECT IS SET WITH
1941                                                                ;RPCA ONE GREATER THAN BOUNDARY
1942                                                                ;SWITCH SETTINGS
1943  011734  012767  011674  167040  2$:   MOV     #3$,LAD
1944  011742  104000                        SCOPE
1945  011744  000241                        CLC
1946  011746  006100                        ROL     R0              ;JPDATE TEST PATTERN
1947  011750  032700  001000                BIT     #B9,R0          ;IS PATTERN EXCEEDED?
1948  011754  001732                        BEQ     4$              ;BRANCH IF NO
1949
1950  011756  000005                  WRSW2:  RESET
1951  011760  012767  000001  001632        MOV     #1,UNIT
1952  011766  004567  167272                JSR     R5,PRINT$       ;PRINT MESSAGE
1953  011772  012603                        MES30                   ;HAVE BOUNDARY SWITCHES CLEARED
1954  011774  004567  167264                JSR     R5,PRINT$       ;PRINT MESSAGE
1955  012000  012641                        MES31
1956  012002                          3$:
1957  012002  004567  167256                JSR     R5,PRINT$       ;PRINT MESSAGE
1958  012006  012571                        MES29
1959  012010  016767  001604  167544        MOV     UNIT,TTY
1960  012016  004767  167334                JSR     PC,PRINTS       ;TYPE LOCATION-SUPRESS ZEROS
1961  012022  004767  167656                JSR     PC,READ$        ;INPUT MESSAGE
1962  012026  012777  000002  001610  2$:   MOV     #2,@RPCA        ;PLACE A 2 IN RPCA
1963  012034  032777  000400  001566        BIT     #BB,@RPDS       ;IS UNIT WRITE PROTECT SET?
1964  012042  001001                        BNE     1$
1965  012044  104400                        HLT                     ;UNIT WRITE PROTECT SHOULD BE SET
1966                                                                ;RPCA CONTAINS A TWO AND
1967                                                                ;SELECTED UNIT IS ZERO
1968  012046  012767  012026  166726  1$:   MOV     #2$,LAD
1969  012054  104000                        SCOPE
1970  012056  000241                        CLC
1971  012060  006167  001534                ROL     UNIT            ;SHIFT UNIT PATTERN
1972  012064  032767  000010  001526        BIT     #B3,UNIT        ;HAS PATTERN BEEN EXCEEDED?
1973  012072  001743                        BEQ     3$
```

```
1974  012074  004567  167164                    JSR     R5,PRINT$       ;PRINT MESSAGE
1975  012100  012726                            MES32
1976  012102  004567  167156                    JSR     R5,PRINT$       ;PRINT MESSAGE
1977  012106  012746                            MES33
1978  012110  000000                            HALT
1979  012112  005015  047510  020127  MES20:    .ASCIZ  <15><12>/HOW MANY DRIVES ARE THERE?/
1980  012120  040515  054516  042040
1981  012126  044522  042526  020123
1982  012134  051101  020105  044124
1983  012142  051105  037505     000
1984  012147     015  042012  051511  MES21:    .ASCIZ  <15><12>/DISABLE DRIVE /
1985  012154  041101  042514  042040
1986  012162  044522  042526  000040
1987  012170  005015  042522  050123  MES22:    .ASCIZ  <15><12>/RESPOND TO ALL REQUEST WITH CR WHEN READY/
1988  012176  047117  020104  047524
1989  012204  040440  046114  051040
1990  012212  050505  042525  052123
1991  012220  053440  052111  020110
1992  012226  051103  053440  042510
1993  012234  020116  042522  042101
1994  012242  000131
1995  012244  005015  047105  041101  MES23:    .ASCIZ  <15><12>/ENABLE DRIVE /
1996  012252  042514  042040  044522
1997  012260  042526  000040
1998  012264  005015  042523  020124  MES24:    .ASCIZ  <15><12>/SET READ ONLY ON DRIVE /
1999  012272  042522  042101  047440
2000  012300  046116  020131  047117
2001  012306  042040  044522  042526
2002  012314  000040
2003  012316  005015  046103  040505  MES25:    .ASCIZ  <15><12>/CLEAR READ ONLY ON DRIVE /
2004  012324  020122  042522  042101
2005  012332  047440  046116  020131
2006  012340  047117  042040  044522
2007  012346  042526  000040
2008  012352  005015  042523  020124  MES26:    .ASCIZ  <15><12>/SET WRITE LOCKOUT AND CLEAR THE BOUNDARY/
2009  012360  051127  052111  020105
2010  012366  047514  045503  052517
2011  012374  020124  047101  020104
2012  012402  046103  040505  020122
2013  012410  044124  020105  047502
2014  012416  047125  040504  054522
2015  012424     000
2016  012425     015  051412  044527  MES27:    .ASCIZ  <15><12>/SWITCHES ON THE RP11C SWITCH PANEL/
2017  012432  041524  042510  020123
2018  012440  047117  052040  042510
2019  012446  051040  030520  041461
2020  012454  051440  044527  041524
2021  012462  020110  040520  042516
2022  012470  000114
2023  012472  005015  046120  041501  MES28:    .ASCIZ  <15><12>/PLACE THE FOLLOWING VALUES IN THE CYLINDER BOUNDRY SWITCHES:/
2024  012500  020105  044124  020105
2025  012506  047506  046114  053517
2026  012514  047111  020107  040526
2027  012522  052514  051505  044440
2028  012530  020116  044124  020105
2029  012536  054503  044514  042116
```

```
 2030  012544  051105  041040  052517
 2031  012552  042116  054522  051440
 2032  012560  044527  041524  042510
 2033  012566  035123     000
 2034  012571     015  044412  051516  MES29:  .ASCIZ  <15><12>/INSERT /
 2035  012576  051105  020124     000
 2036  012603     015  041412  042514  MES30:  .ASCIZ  <15><12>/CLEAR THE BOUNDARY SWITCHES/
 2037  012610  051101  052040  042510
 2039  012616  041040  052517  042116
 2039  012624  051101  020131  053523
 2040  012632  052111  044103  051505
 2041  012640     000
 2042  012641     015  051412  052105  MES31:  .ASCIZ  <15><12>/SET FOLLOWING VALUE IN THE DRIVE BOUNDARY SWITCHES/
 2043  012646  043040  046117  047514
 2044  012654  044527  043516  053040
 2045  012662  046101  042525  044440
 2046  012670  020116  044124  020105
 2047  012676  051104  053111  020105
 2048  012704  047502  047125  040504
 2049  012712  054522  051440  044527
 2050  012720  041524  042510  000123
 2051  012726  005015  042524  052123  MES32:  .ASCIZ  <15><12>/TEST COMPLETE/
 2052  012734  041440  046517  046120
 2053  012742  052105  000105
 2054  012746  005015  042522  042523  MES33:  .ASCIZ  <15><12>/RESET WRITE LOCKOUT....THANK YOU/
 2055  012754  020124  051127  052111
 2056  012762  020105  047514  045503
 2057  012770  052517  027124  027056
 2058  012776  052056  040510  045516
 2059  013004  054440  052517     000
 2060
 2061          013012                  .EVEN
```

```
                                        .SBTTL  *** SUBROUTINES ***
                                        ;FLOAT A ONE AND ZERO THRU A DESIGNATED REGISTER
013012  012567  000576          BITST:  MOV     (R5)+,MASK      ;FETCH DATA MASK
013016  012504                          MOV     (R5)+,R4        ;GET REGISTER UNDER TEST
013020  010567  000602                  MOV     R5,LERR         ;GET ERROR RETURN ADDR
013024  062705  000004                  ADD     #4,R5           ;MODIFY RETURN ADDR.
013030  012703  000001                  MOV     #1,R3           ;INITIALIZE DATA PATTERN
013034  004767  000016          BTL1:   JSR     PC,BTL2         ;OUTPUT FLOATING ZERO
013040  004767  000012                  JSR     PC,BTL2         ;OUTPUT FLOATING ONE
013044  000241                          CLC
013046  006103                          ROL     R3              ;SHIFT PATTERN
013050  005703                          TST     R3
013052  001370                          BNE     BTL1
013054  000205                          RTS     R5
013056  005103                  BTL2:   COM     R3              ;COMPLIMENT PATTERN
013060  010367  000420          BTL3:   MOV     R3,EXP$
013064  005167  000524                  COM     MASK            ;AND MASK WITH PATTERN
013070  046767  000520  000406          BIC     MASK,EXP$
013076  005167  000512                  COM     MASK
013102  016714  000376                  MOV     EXP$,(R4)       ;OUTPUT TO REGISTER
013106  011467  000374                  MOV     (R4),REC$       ;INPUT FROM REGISTER
013112  005167  000476                  COM     MASK
013116  046767  000472  000362          BIC     MASK,REC$       ;AND MASK AND RECEIVED DATA
013124  005167  000464                  COM     MASK
013130  026767  000350  000350          CMP     EXP$,REC$       ;IS DATA CORRECT
013136  001402                          BEQ     BTN2
013140  004777  000462                  JSR     PC,@LERR        ;GO REPORT THE ERROR
013144  012767  013060  165630  BTN2:   MOV     #BTL3,LAD
013152  104000                          SCOPE
013154  000207                          RTS     PC

                                        ;RAPID DATA TEST FOR W.C. AND B.A. REGISTERS

013156  012501                  RAPBIS: MOV     (R5)+,R1        ;GET REGISTER ADDRESS IN R1
013160  012567  000442                  MOV     (R5)+,LERR      ;GET ERROR RETURN ADDRESS
013164  005725                          TST     (R5)+           ;MODIFY RETURN ADDRESS
013166  012737  013234  000004          MOV     #2$,@#ERRVEC    ;SETUP FOR END MEMORY TRAP
013174  005000                          CLR     R0              ;SETUP START MEMORY LOCATION
013176  011011                  1$:     MOV     (R0),(R1)       ;MOVE MEMORY TO REGISTER
013200  021120                          CMP     (R1),(R0)+      ;TEST FOR PROPER VALUE IN REGISTER
013202  001775                          BEQ     1$
013204  014067  000274                  MOV     -(R0),EXP$      ;MOVE GOOD DATA TO EXP$ FOR TYPOUT
013210  011167  000272                  MOV     (R1),REC$       ;MOVE REGISTER DATA TO REC$ FOR ERROR TYPOUT
013214  004777  000406                  JSR     PC,@LERR        ;REPORT ERROR
013220  012767  013176  165554          MOV     #1$,LAD         ;LOOP ADDRESS FOR SCOPE
013226  104000                          SCOPE
013230  005720                          TST     (R0)+           ;NO LOOP CONTINUE TEST
013232  000761                          BR      1$
013234  022626                  2$:     CMP     (SP)+,(SP)+     ;CLEAN OFF STACK
013236  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC      ;RESTORE TRAP CATCHER
013244  012737  000340  177776          MOV     #PRI7,@#PSW     ;RESTORE NO INTERUPTS
013252  000205                          RTS     R5              ;RETURN TO CALLER

                                        ;INITALIZE TRAP VECTORS AND LOCKOUT TRAPS
```

```
2118
2119  013254  012767  001112  164552  INIT:   MOV     #ERROR,34       ;SETUP TRAP VECTOR
2120  013262  012767  000340  164546          MOV     #PRI7,36
2121  013270  012767  001004  164532          MOV     #SCOPE$,30      ;SETUP EMT VECTOR
2122  013276  012767  000340  164526          MOV     #PRI7,32
2123  013304  012737  000340  177776          MOV     #PRI7,@#PSW     ;LOCKOUT INTERRUPTS
2124  013312  000207                          RTS     PC
2125
2126                                  ;TYPE ERROR MESSAGES
2127
2128  013314  032767  000002  165720  MSG:    BIT     #B1,HLTCT$      ;TYPE ENTIRE MESSAGE?
2129  013322  001033                          BNE     1$              ;BRANCH IF NO
2130  013324  004567  165734                  JSR     R5,PRINT$       ;PRINT MESSAGE
2131  013330  013740                          MES8
2132  013332  004567  165726                  JSR     R5,PRINT$       ;PRINT MESSAGE
2133  013336  013727                          MES2A
2134  013340  017767  000264  166214          MOV     @RPDS,TTY
2135  013346  004767  165772                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
2136  013352  004567  165706                  JSR     R5,PRINT$       ;PRINT MESSAGE
2137  013356  013705                          MES1A
2138  013360  017767  000246  166174          MOV     @RPER,TTY
2139  013366  004767  165752                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
2140  013372  004567  165666                  JSR     R5,PRINT$       ;PRINT MESSAGE
2141  013376  013716                          MES2
2142  013400  017767  000230  166154          MOV     @RPCS,TTY
2143  013406  004767  165732                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
2144  013412  032767  000001  165622  1$:     BIT     #B0,HLTCT$      ;TYPE EXPECTED RECEIVED?
2145  013420  001001                          BNE     2$              ;BRANCH IF YES
2146  013422  000207                          RTS     PC
2147  013424  032767  000002  165610  2$:     BIT     #B1,HLTCT$
2148  013432  001403                          BEQ     3$
2149  013434  004567  165624                  JSR     R5,PRINT$       ;PRINT MESSAGE
2150  013440  014000                          MES17
2151  013442                          3$:
2152  013442  004567  165616                  JSR     R5,PRINT$       ;PRINT MESSAGE
2153  013446  014020                          MES18
2154  013450  016767  000030  166104          MOV     EXP$,TTY
2155  013456  004767  165662                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
2156  013462  004567  165576                  JSR     R5,PRINT$       ;PRINT MESSAGE
2157  013466  014033                          MES19
2158  013470  016767  000012  166064          MOV     REC$,TTY
2159  013476  004767  165642                  JSR     PC,PRINTR       ;TYPE LOCATION WITH LEADING ZEROS
2160  013502  000207                          RTS     PC
2161  013504  000000                  EXP$:   0
2162  013506  000000                  REC$:   0
2163
2164
2165                                  ;ROUTINE TO CLEAR THE RP11C IN MAINT MODE
2166
2167  013510  012700  000003          CLRP:   MOV     #3,R0
2168  013514  012777  000001  000112          MOV     #1,@RPCS
2169  013522  012767  060001  000074          MOV     #B14+B13+B0,TPL
2170  013530  004767  000010                  JSR     PC,TIMEP        ;GENERATE 2 TIME PULSES
2171  013534  052777  020000  000116          BIS     #B13,@RPM3      ;SET UNIT ON LINE
2172  013542  000207                          RTS     PC
2173
```

```
                                        ;GENERATE TIMMING PULSES
                                        
013544  016777  000054  000106  TIMEP:  MOV     TPL,@RPM3       ;SET SPECIFIED MAINT BITS
013552  005300                          DEC     R0
013554  005700                          TST     R0              ;DONE?
013556  001372                          BNE     TIMEP           ;0=YES
013560  000207                          RTS     PC
                                        
013562  012700  000003          T3P:    MOV     #3,R0
013566  000167  177752                  JMP     TIMEP           ;GENERATE 3 TIMING PULSES
                                        
                                        ;GENERATE AN INDEX PULSE
                                        
013572  052777  010000  000060  INDEXP: BIS     #B12,@RPM3      ;GENERATE AN INDEX PULSE
013600  042777  010000  000052          BIC     #B12,@RPM3
013606  005300                          DEC     R0
013610  001370                          BNE     INDEXP          ;DONE?
013612  000207                          RTS     PC
                                        
                                        
013614  000000                  MASK:   0
013616  000000                  PASSCT: 0
013620  000000                  UNIT:   0
013622  000000                  HUNIT:  0
013624  000000                  TPL:    0
013626  000000                  LERR:   0
013630  176710                  RPDS:   176710
013632  176712                  RPER:   176712
013634  176714                  RPCS:   176714
013636  176715                  RPCS1:  176715
013640  176716                  RPWC:   176716
013642  176720                  RPBA:   176720
013644  176722                  RPCA:   176722
013646  176724                  RPDA:   176724
013650  176725                  RPDA1:  176725
013652  176726                  RPM1:   176726
013654  176730                  RPM2:   176730
013656  176731                  RPM21:  176731
013660  176732                  RPM3:   176732
013662  176736                  RPS0:   176736
013664  176734                  SUCA:   176734
                                        
                                        ;MESSAGES USED BY THE PROGRAM
                                        
013666  005015  047105  020104  MES1:   .ASCIZ  <15><12>/END OF PASS /
013674  043117  050040  051501
013702  020123     000
013705     015  051012  042520  MES1A:  .ASCIZ  <15><12>/RPER= /
013712  036522  000040
013716  005015  050122  051503  MES2:   .ASCIZ  <15><12>/RPCS= /
013724  020075     000
013727     015  051012  042120  MES2A:  .ASCIZ  <15><12>/RPDS= /
013734  036523  000040
013740  005015  052123  052101  MES8:   .ASCIZ  <15><12>/STATUS ERROR/
013746  051525  042440  051122
013754  051117     000
```

```
 2230
 2231  013757  000015  052012  051505  MES10:  .ASCIZ  <15><12>/TEST PATTERN= /
 2232  013764  020124  040520  052124
 2233  013772  051105  036516  000040
 2234  014000  005015  047503  050115  MES17:  .ASCIZ  <15><12>/COMPARE ERROR/
 2235  014006  051101  020105  051105
 2236  014014  047522  000122
 2237
 2238  014020  005015  054105  042520  MES18:  .ASCIZ  <15><12>/EXPECTED/
 2239  014026  052103  042105  000000
 2240  014033  000015  051012  041505  MES19:  .ASCIZ  <15><12>/RECEIVED /
 2241  014040  044505  042526  020104
 2242  014046  000000
 2243          014050                          .EVEN
 2244          000001                          .END
```

```
BABT    002654          797#
BACK  = 002336          713#    716
BELL    001250          512     533#
BITST   013012          732     758     780     798     816     833    2066#
BTL1    013034         2071#   2076
BTL2    013056         2071    2072    2078#
BTL3    013060         2079#   2091
BTN2    013144         2089    2091#
B0    = 000001          387#    921    1042    1066    1090    1112    1145    1168    1200    1222    1250    1273    1486
                       1667    1754    1757    1787    2144    2169
B1    = 000002          386#   1760    2128    2147
B1C   = 002000          377#    916     939     940    1252    1685    1707    1736    1760    1787
B11   = 004000          376#    509     917     941     951     962     963    1063    1066    1202    1224    1572    1611
                       1619
B12   = 010000          375#   1147    1170    1580    1624    1636    2187    2188
B13   = 020000          374#    513     539     557     560    1042    1066    1090    1092    1112    1114    1145    1168
                       1200    1222    1250    1273    1300    1314    1331    1343    1486    1667    1754    1757    2169
                       2171
B14   = 040000          373#    473     891     901     902     925    1042    1049    1066    1068    1072    1090    1112
                       1122    1145    1168    1178    1200    1222    1232    1250    1256    1273    1279    1486    1667
                       1754    1757    1839    1851    2169
B15   = 100000          372#    526     892     903     929     974     985     986    1039    1042    1044    1053    1076
                       1126    1182    1236    1260    1283    1601
B2    = 000004          385#   1707
B3    = 000010          384#   1685    1972
B4    = 000020          383#   1787
B5    = 000040          382#   1760    1787
B6    = 000100          381#   1360    1390    1406    1421    1436    1452    1468    1736
B7    = 000200          380#    766     771
B8    = 000400          379#    975     987    1014    1275    1321    1508    1516    1517    1518    1519    1522    1523
                       1542    1543    1754    1780    1781    1782    1871    1883    1906    1911    1932    1938    1963
B9    = 001000          378#    952     964    1947
C     = 000001          337#
CABT    002732          815#
CAF1    007100         1496#
CAL1    007106         1498#   1505    1509
CLRP    013510          877    1085    1107    1140    1163    1195    1217    1248    1269    1298    1329    1389    1404
                       1419    1434    1450    1466    1480    1498    1513    1538    1568    1609    2167#
CSBT    002502          758#
CSF1    005706         1294#
CSF10   006672         1450#   1461
CSF11   006754         1466#   1475
CSF12   007022         1480#   1491
CSF2    006234         1357#   1367
CSF3    006312         1372#   1383
CSF4    006370         1388#   1399
CSF5    006450         1404#   1414
CSF6    006530         1419#   1429
CSF7    006610         1434#   1445
DABT    003010          833#
DAF1    007164         1513#   1533
DAF2    007312         1538#   1553
DISPLA= 177570          366#
DSF1    003210          889#    895
DSF10   003526          960#    967
DSF11   003574          972#    978
```

```
DSF12   003634          983#   990
DSF13   003702          995#
DSF2    003250          900#   907
DSF4    003312          913#   932
DSF5    003420          937#   944
DSF7    003466          949#   955
EMTVEC= 000030          356#
ERBT    002356          731#  1799
ERF1    004136         1038#  1056
ERF10   005322         1217#  1239  1243
ERF11   005474         1247#  1264
ERF12   005576         1269#  1286
ERF2    004246         1061#  1079
ERF3    004362         1084#
ERF4    004510         1107#  1129  1133
ERF5    004662         1137#
ERF6    005016         1163#  1185  1189
ERF7    005170         1193#
ERGO    002434          746#   753
ERL5    004674         1140#  1154  1158
ERL7    005200         1195#  1209  1213
ERRFLG  001240          475    477*   526*  529#
ERROR   001112          506#  2119
ERRORS  001246          527*   532#
ERRPC   001252          516    534#
ERRVEC= 000004          351#  2100*  2113*
EXP$    013504          742*   751*   768*  790*  808*  826*  843*  881*  995*  999  1000  1006  1013*
                       1014   1028*  1084*  1087  1097  1101*  1102  1109  1119  1131*  1132  1138*  1142
                       1152   1156*  1157   1165  1175  1187*  1188  1194*  1197  1207  1211*  1212  1219
                       1229   1241*  1242   1294*  1301  1309  1313  1320*  1321  1497*  1499  1501  1507*
                       1508   1539*  1550   1555*  1556  1569*  1576  1585  1593*  1597*  1600*  1601*  1610*
                       1615   1616*  1617   1623*  1629  1633*  1634  1655*  1672*  1675*  1676  1685*  1686
                       1694*  1697*  1698   1707*  1708  1728*  1736*  1737  1760*  1761  1787*  1788  2079*
                       2081*  2083   2088   2105*  2154  2161#
HINUM   001700          601    611    622*   626#
HLT   = 104400          413#   735    743    752   761   769   773   793   791   801   809   819   827
                        836    844    854    866   882   894   905   919   923   927   931   943   954
                        966    977    989   1003  1009  1020  1031  1046  1051  1055  1070  1074  1078
                       1094   1116   1124   1128  1149  1172  1180  1184  1204  1226  1234  1238  1254
                       1258   1262   1277   1281  1285  1306  1316  1337  1349  1365  1382  1397  1412
                       1427   1444   1460   1474  1484  1490  1503  1532  1552  1574  1582  1587  1613
                       1621   1626   1631   1638  1656  1678  1688  1700  1710  1729  1739  1763  1790
                       1841   1853   1873   1885  1908  1913  1934  1940  1965
HLTAD$  001244          517*   518*   519    520   531#
HLTCT$  001242          519*   530#  2128   2144  2147
HUNIT   013622         1827*  1857   1889   2197#
ICNT    001000          463#   721*  1796*  1797
INDEXP  013572         1515   1521   1541   1779  2187#  2190
INIT    013254          722   1816   2119#
INPUT$  002014          629    630    653#   664   705
IOTVEC= 000020          354#
KIPAR0= 172340          392#
KIPAR1= 172342          393#
KIPAR2= 172344          394#
KIPAR3= 172346          395#
KIPAR4= 172350          396#
```

```
KIPAR5= 172352            397#
KIPAR6= 172354            398#
KIPAR7= 172356            399#
KIPDR0= 172300            400#
KIPDR1= 172302            401#
KIPDR2= 172304            402#
KIPDR3= 172306            403#
KIPDR4= 172310            404#
KIPDR5= 172312            405#
KIPDR6= 172314            406#
KIPDR7= 172316            407#
LAC     001002            464#   479    744#   753#   774#   792#   810#   828#   845#   857#   869#   883#   895#
                          907#   932#   944#   955#   967#   978#   990#  1004#  1010#  1021#  1032#  1056#  1079#
                         1099#  1129#  1154#  1185#  1209#  1239#  1264#  1286#  1318#  1351#  1367#  1393#  1399#
                         1414#  1429#  1445#  1461#  1475#  1491#  1505#  1533#  1553#  1588#  1640#  1713#  1743#
                         1767#  1794#  1855#  1887#  1915#  1943#  1968#  2091#  2108#
LERR    013626           2068#  2090   2098#  2107   2199#
LONUM   001702            600    609    621#   627#
LOOP    002346            721#  1805
MASK    013614           2066#  2080#  2081   2082#  2085#  2086   2087#  2194#
MES1    013666           1802   2218#
MES1A   013705           2137   2221#
MES10   013757           1096   1118   1151   1174   1206   1228   1308   2231#
MES17   014000           2150   2234#
MES18   014020           2153   2238#
MES19   014033           2157   2240#
MES2    013716           2141   2223#
MES2A   013727           2133   2225#
MES20   012112           1819   1979#
MES21   012147           1833   1984#
MES22   012170           1830   1987#
MES23   012244           1845   1995#
MES24   012264           1865   1998#
MES25   012316           1877   2003#
MES26   012352           1901   2008#
MES27   012425           1903   2016#
MES28   012472           1920   2023#
MES29   012571           1923   1958   2034#
MES30   012603           1953   2036#
MES31   012641           1955   2042#
MES32   012726           1975   2051#
MES33   012746           1977   2054#
MES8    013740           2131   2227#
MSG     013314            522   2128#
N     = 000010            340#
NUM$    002334            662#   670#   676#   682#   687#   695#   711#  1822   1824#  1825   1827
PACK$   002062            660#  1821
PAC$    002302            669    671    677    683    688    702#
PASSCT  013616            720#  1800#  1803   2195#
PC    =%000007            334#   461#   490#   501#   521#   522#   559#   562#   595#   599#   623#   624#   651#
                          661#   669#   671#   677#   683#   688#   699#   700#   708#   715#   722#   736#   762#
                          784#   802#   820#   837#   855#   867#   877#  1043#  1067#  1085#  1091#  1098#  1107#
                         1113#  1120#  1140#  1146#  1153#  1163#  1169#  1176#  1195#  1201#  1208#  1217#  1223#
                         1230#  1248#  1251#  1269#  1274#  1298#  1310#  1329#  1388#  1404#  1419#  1434#  1450#
                         1466#  1480#  1487#  1498#  1513#  1515#  1521#  1538#  1541#  1568#  1609#  1669#  1682#
                         1691#  1704#  1733#  1753#  1756#  1777#  1779#  1784#  1804#  1816#  1820#  1821#  1835#
```

```
                    1836*  1847*  1848*  1867*  1868*  1879*  1880*  1904*  1927*  1930*  1960*  1961*  2071*
                    2072*  2090*  2093*  2107*  2124*  2135*  2139*  2143*  2146*  2155*  2159*  2160*  2170*
                    2172*  2180*  2191*
PFVEC = 000024       355#
PKEX$   002270       697#   703
PKEX1$  002274       696    698#
PK$     002332       670    673*   674*   675*   676    679*   680*   681*   682    684*   686*   687    689*
                     691*   692*   693*   694*   695    704*   705*   707*   710#
PRINTA  001370       558    563#
PRINTB  001400       561    565#
PRINTR  001344       521    557#   1098   1120   1153   1176   1208   1230   1310   2135   2139   2143   2155
                    2159
PRINTS  001356       560#   1804   1835   1847   1867   1879   1927   1960
PRINT$  001264       515    539#   592    639    648    1095   1117   1150   1173   1205   1227   1307   1801
                    1818   1829   1832   1844   1864   1876   1900   1902   1919   1922   1952   1954   1957
                    1974   1976   2130   2132   2136   2140   2149   2152   2156
PRI1  = 000040       348#   1359
PRI2  = 000100       347#   1391
PRI3  = 000140       346#   1407
PRI4  = 000200       345#   1422
PRI5  = 000240       344#   1437
PRI6  = 000300       343#   1453
PRI7  = 000340       342#   1296   1305   1327   1336   1348   1364   1379   1396   1411   1426   1441   1457
                    2114   2120   2122   2123
PRNTF$  001302       511    540    543#
PSW   = 177776       360#   1299*  1305*  1330*  1336*  1342*  1348*  1359*  1364*  1375*  1379*  1391*  1396*
                    1407*  1411*  1422*  1426*  1437*  1441*  1453*  1457*  2114*  2123*
RAND$   001564       598#
RAPBA   003116       864#   869
RAPBIS  013156       852    864    2097#
RAPWC   003070       852#   857
RDSW    011350      1858   1861#
READL$  002060       649    655#
READM$  002054       640    654#
READT   010550      1749#
READ$   001704       628#   1820   1836   1848   1868   1880   1904   1930   1961
REC$    013506       741*   748*   749    765*   766*   789*   807*   825*   840*   841*   878*   879    1002*
                    1029*  1030*  1500*  1501   1544*  1545*  1546*  1547*  1548*  1549*  1550   1584*  1585
                    1628*  1629   1652*  1653*  1670*  1671*  1676   1683*  1684*  1686   1692*  1693*  1698
                    1705*  1706*  1708   1725*  1726*  1734*  1735*  1737   1758*  1759*  1761   1785*  1786*
                    1788   2084*  2086*  2088   2106*  2158   2162#
RESTOR  010434      1724#
REST$   001064       494#   623    699
RESVEC= 000010       352#
RESWC   003144       876#   883
RPBA    013642      2205#
RPBT    002576       779#
RPCA    013644      1087*  1109*  1658*  1661*  1905*  1910*  1931*  1937*  1962*  2206#
RPCS    013634       747*   763*   765    771    925    929    1041*  1049   1053   1065*  1072   1076   1099*
                    1111*  1122   1126   1144*  1167*  1178   1182   1199*  1221*  1232   1236   1249*  1256
                    1260   1272*  1279   1283   1300*  1314   1331*  1343*  1360*  1390*  1406*  1421*  1436*
                    1452*  1468*  1481*  1482   1498   1666*  1731*  1751*  1775*  1837*  1849*  1869*  1981*
                    2142   2168*  2202#
RPCS1   013636      1838*  1850*  1870*  1882*  2203#
RPDA    013646       838*   840    1139*  1193*  1197*  1219*  1524   1544   1774*  2207#
RPDA1   013650      1142*  1165*  1662*  1665*  2208#
```

```
RPDS    013630   890* 892  903  915* 917  938* 941  950* 952  961* 964  973* 975
                 984* 987  998* 1000 1002 1006* 1007 1018 1025* 1026 1030 1313* 1344*
                 1839 1851 1871 1883 1906 1911 1932 1938 1963 2134 2200#
RPER    013632   746* 748  921  1044 1062* 1068 1086* 1092 1108* 1114 1141* 1147 1164*
                 1170 1196* 1202 1218* 1224 1247* 1252 1270* 1275 2138 2201#
RPM1    013652   1572 1580 1611 1619 1624 1636 1652 1670 1683 1692 1705 1725 1734
                 1758 1785 2209#
RPM2    013654   996* 999* 1016* 1023* 1024* 1301* 1332* 1374* 2210#
RPM21   013656   1499* 2211#
RPM3    013660   889* 891* 900* 901* 902* 914* 916* 937* 939* 940* 949* 951* 960*
                 962* 963* 972* 974* 983* 985* 986* 997* 1039* 1063* 1295* 1516* 1517*
                 1518* 1519* 1522* 1523* 1542* 1543* 1780* 1781* 1782* 2171* 2176* 2187* 2188*
                 2212#
RPS0    013662   1576* 1584 1615* 1628 2213#
RPWC    013640   876* 878  1040* 1064* 1088* 1110* 1143* 1166* 1198* 1220* 1271* 1750* 1773*
                 2204#
RW    = 000006   408#
R0    =%000000   327# 489  495* 600* 604* 613* 619* 621  663* 664  666* 668*  702
                 705  706* 1514* 1520* 1524* 1525* 1526* 1527* 1528* 1529* 1530 1540* 1570*
                 1590 1596* 1668* 1681* 1690* 1703* 1732* 1752* 1755* 1776* 1778* 1783* 1899*
                 1925* 1926 1929* 1931 1946* 1947 2101* 2102 2103 2105 2110 2167* 2177*
                 2178 2182* 2189*
R1    =%000001   328# 488  496* 601* 605* 610* 611* 614* 616* 620* 622  1577* 1578*
                 2097* 2102* 2103 2106
R2    =%000002   329# 487  497* 603* 606* 609* 612* 615* 617* 618* 619
R3    =%000003   330# 486  498* 602* 607* 628  629* 630  650* 2070* 2074* 2075 2078*
                 2079
R4    =%000004   331# 485  499* 567  568* 569* 571* 588  594* 1567* 1642* 1651* 1659
                 1663 1673 1695 1715 1717* 2067* 2083* 2084
R5    =%000005   332# 484  500* 511* 515* 541* 542* 545  547* 548  550* 551*  552
                 592* 639* 648* 732* 758* 780* 798* 816* 833* 852* 864* 1095* 1117*
                 1150* 1173* 1205* 1227* 1307* 1801* 1818* 1829* 1832* 1844* 1864* 1876* 1900*
                 1902* 1919* 1922* 1952* 1954* 1957* 1974* 1976* 2066 2067 2068 2069* 2077*
                 2097 2098 2099 2115* 2130* 2132* 2136* 2140* 2149* 2152* 2156*
SAVE$   001036   483# 599  661
SCOPE = 104000   415# 745  754  775  793  811  829  846  858  870  884  896  908
                 933  945  956  968  979  991  1005 1011 1022 1033 1057 1080 1100
                 1130 1155 1186 1210 1240 1265 1287 1319 1352 1368 1384 1400 1415
                 1430 1446 1462 1476 1492 1506 1534 1554 1589 1641 1714 1744 1768
                 1795 1856 1888 1916 1944 1969 2092 2109
SCOPE$  001004   473# 2121
SEEK    010026   1643 1650#
SEEK1   010432   1716 1719#
SILO    007440   1567#
SILOR   007444   1568# 1644
SILO1   007630   1609# 1640
SP    =%000006   333# 479*  483  484* 485* 486* 487* 488* 489* 494  495  496  497
                 498  499  500  545* 546* 550  567* 594  697* 719* 1312* 1340* 1350*
                 1366* 1381* 1398* 1413* 1428* 1443* 1459* 1473* 1815* 2112
SR0   = 177572   391#
START   002336   461  719#
STATUS= 000256   718# 1296* 1327*
STKPTR= 000500   369# 719  1312 1340 1350 1366 1381 1398 1413 1428 1443 1459 1473
                 1815
SLCA    013664   1500 2214#
SWR   = 177570   365# 473  506  509  513  523  539  557  560
```

```
SWTST   011102            715    1815#
T       = 000020          341#
TBITVE= 000014            353#
TIMEP   013544           1669    1682    1691    1704    1733    1753    1756    1777    1784    2170    2176#   2179    2183
TKB     = 177562          362#     642
TKS     = 177560          361#
TPB     = 177566          364#     552*    642*
TPL     013624           1042*   1066*   1090*   1112*   1145*   1168*   1200*   1222*   1250*   1273*   1486*   1667*   1754*
                         1757*   2169*   2176    2198#
TPS     = 177564          363#     543     553     643
TRAPVE= 000034            357#
TSTAT   006070           1327#   1351
TTY     001562            520*    574*    576*    578*    597#   1097*   1119*   1152*   1175*   1207*   1229*   1309*   1803*
                         1834*   1846*   1866*   1878*   1926*   1959*   2134*   2138*   2142*   2154*   2158*
T3P     013562           1043    1067    1091    1113    1146    1169    1201    1223    1251    1274    1487    2182#
UNIT    013620           1828*   1834    1838    1846    1850    1857    1859*   1862*   1866    1870    1878    1882    1989
                         1891*   1893*   1951*   1959    1971*   1972    2196#
UP      = 000000          409#
V       = 000002          338#
VECTOR= 000254            717#   1297*   1328*   1341*   1356*   1373*   1389*   1405*   1420*   1435*   1451*   1467*
WRITE   010670           1773#   1794
WRSW    011536           1898#
WRSW1   011634           1918#
WRSW2   011756           1950#
Z       = 000004          339#
.       = 014050          459#    460#    462#    536#    544     554     564     581     584     589     596#    633     644
                          653#    713     714#    716#   2061#   2243#
.PR     001536            563*    565*    566*    568     572     582*    583     586*    588     593     596#
.PRF    001452            570     573     579#
.PRL    001424            571#    587
.PTT    001412            567#
```

BITEST	719#	731	779	797	815										
DUMP	451#	520	1097	1119	1152	1175	1207	1229	1309	2134	2138	2142	2154	2158	
DUMPF	449#														
LOPDR	411#														
PACK	457#	1821													
PRINT	446#	515	592	638	648	1095	1117	1150	1173	1205	1227	1307	1801	1817	1829
	1831	1843	1863	1875	1900	1902	1918	1921	1952	1954	1956	1974	1976	2130	2132
	2136	2140	2149	2151	2156										
PRINTF	447#	511													
RAND	454#														
READ	455#	1820	1836	1848	1868	1880	1904	1930	1961						
REST	443#	623	698												
SAVE	442#	598	660												
SDUMP	452#	1803	1834	1846	1866	1878	1926	1959							
SDUMPF	450#														
.DUMP	448#	556													
.ERROR	444#	503													
.PACK	456#	657													
.PRINT	445#	537													
.RAND	453#	598													
.READ	454#	628													
.REST	442#	492													
.SAVE	442#	481													
.SCOPE	442#	465													

```
        1398  1404  1419  1434  1450  1466  1480  1487  1498  1513  1515  1521  1538  1541  1568
        1609  1669  1682  1691  1704  1733  1753  1756  1777  1779  1784  1801  1804  1816  1818
        1820  1821  1829  1832  1835  1836  1944  1847  1848  1864  1967  1868  1876  1879  1880
        1900  1902  1904  1919  1922  1927  1930  1952  1954  1957  1960  1961  1974  1976  2071
        2072  2090  2107  2130  2132  2135  2136  2139  2140  2143  2149  2152  2155  2156  2159
        2170
MOV      461   479   483   484   485   486   487   488   489   490   494   495   496   497   498
         499   500   501   517   520   545   547   550   567   568   594   600   601   602   621
         622   628   629   642   650   670   715   719   737   741   744   746   747   748   753
         763   765   774   785   789   792   803   807   810   821   825   828   838   840   845
         857   869   876   878   883   895   907   932   944   955   967   978   990   995   998
         999  1004  1006  1010  1016  1021  1024  1032  1040  1042  1056  1064  1066  1079  1087
        1088  1089  1090  1097  1099  1109  1110  1111  1112  1119  1129  1143  1144  1145  1152
        1154  1166  1167  1168  1175  1185  1197  1198  1199  1200  1207  1209  1219  1220  1221
        1222  1229  1239  1249  1250  1264  1271  1272  1273  1286  1294  1296  1297  1301  1305
        1309  1312  1313  1318  1327  1328  1332  1336  1340  1341  1344  1348  1350  1351  1358
        1359  1364  1366  1367  1373  1379  1381  1383  1389  1391  1396  1398  1399  1405  1407
        1411  1413  1414  1420  1422  1426  1428  1429  1435  1437  1441  1443  1445  1451  1453
        1457  1459  1461  1467  1473  1475  1481  1486  1491  1500  1505  1514  1520  1524  1533
        1540  1544  1553  1567  1569  1576  1577  1584  1588  1596  1597  1610  1615  1623  1628
        1640  1652  1658  1661  1666  1667  1668  1670  1672  1675  1681  1683  1685  1690  1692
        1694  1697  1703  1705  1707  1713  1725  1731  1732  1734  1736  1743  1750  1751  1752
        1754  1755  1757  1758  1760  1767  1773  1774  1775  1776  1778  1783  1785  1787  1794
        1803  1815  1827  1834  1837  1846  1849  1855  1866  1869  1878  1881  1887  1899  1910
        1915  1926  1931  1943  1951  1959  1962  1968  2066  2067  2068  2070  2079  2083  2084
        2091  2097  2098  2100  2102  2105  2106  2108  2113  2114  2119  2120  2121  2122  2123
        2134  2138  2142  2154  2158  2167  2168  2169  2176  2182
MOVB     519   552   563   566   590   634   705  1002  1030  1041  1065  1142  1165  1374  1499
        1662  1665  1838  1850  1870  1882
NOP     1302  1303  1304  1333  1334  1335  1345  1346  1347  1361  1362  1363  1376  1377  1379
        1393  1394  1395  1408  1409  1410  1423  1424  1425  1438  1439  1440  1454  1455  1456
        1469  1470  1471  1719
RESET    738   764   786   804   822   839   913  1017  1038  1061  1137  1357  1372  1496  1571
        1650  1712  1724  1742  1749  1766  1793  1861  1898  1950
ROL      574   576   578   605   606   673   674   675   686   691   692   693   694  1013  1320
        1593  1929  1946  1971  2074
ROLB     575   577   579
ROR      680   681  1525  1526  1527  1528  1545  1546  1547  1548  1925
RTI      478   480   528
RTS      542   551   559   562   595   624   651   700   708   736   762   784   802   820   937
         855   867  2077  2093  2115  2124  2146  2160  2172  2180  2191
SUB      518  1824
SWAB     679   684   689
TRAP     413
TST      475   506   523   702   739   749   787   805   823   879  1530  1590  1659  1663  1673
        1695  1715  1822  2075  2099  2110  2178
TSTB     543   548   553   580   583   632   643   664  1007  1018  1482  1498
.ABS     320
.ASCIZ   533   534   654   655  1979  1984  1987  1995  1998  2003  2008  2016  2023  2034  2036
        2042  2051  2054  2218  2221  2223  2225  2227  2231  2234  2238  2240
.BLKW    596   653
.END    2244
.ENDC    523
.EVEN    536  2061  2243
.IF      522
.LIST    318   459
```

```
.MACR     411     442     443     444     445     446     447     448     449     450     451     453     454     455     456
          457     719
.MACRO    452
.NLIST    319     459
.PAGE     712     885    1034    1289    1494    1511    1559    1806    2062
.REM       14
.REPT     459
.SBTTL    885    1034    1290    1494    1511    1559    1806    2062
.TITLE    321     712
.WORD     733     734     759     760     781     782     799     800     817     818     834     835     853     865
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZRPAD.SEQ/SOL/CRF/PAGNUM=DZRPAD.MO1,DZRPAD.P11
RUN-TIME:  8 14 3 SECONDS
RUN-TIME RATIO:  141/26=5.2
CORE USED:  8K  (15 PAGES)
```

G05

Spooler runtime 7 Seconds, 33 KCS, 220 disk reads, 3 disk writes, 57 pages