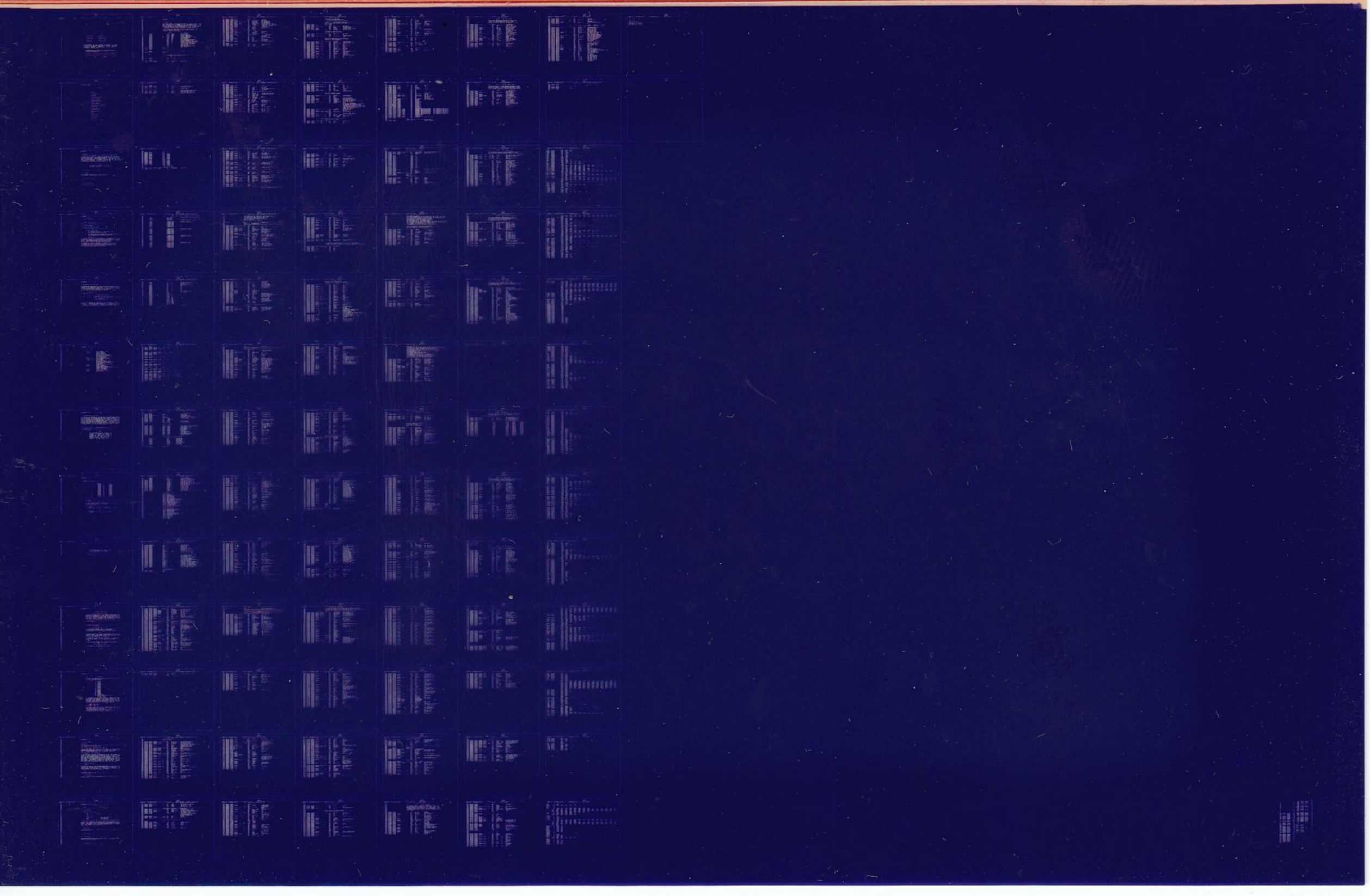
RH11,RS03

DATA RELIABILITY
CZRSCGO

AH-9319G-MC

FICHE 1 OF 1

APR 1978
digital
MADE IN USA



IDENTIFICATION

PRODUCT CODE: AC-9318G-MC
PRODUCT NAME: CZRSCGO RH11-RS03 4 DA RELIAB
PROGRAM DATE: 1-FEBRUARY-1978
MAINTAINER: DIAGNOSTIC GROUP
MODIFIED BY: BILL SCHLITZKUS

TABLE OF CONTENTS

CONTENTS

DESCRIPTION

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03, RS03/LA AND RS04 DRIVES.

THE DZRSC DISK DATA TEST IS A SERIES OF ADDRESS AND DATA RELIABILITY ROUTINES WHICH VERIFY TO THE USER THAT THE CONTROLLER (RH11) AND THE DISKS (RS03/LA OR RS04) ARE OPERATING CORRECTLY. THIS TEST SHOULD BE USED IN CONJUNCTION WITH THE DZRSB DIAGNOSTIC. IF THERE IS A POWER FAIL WHILE THE DIAGNOSTIC IS RUNNING, THE PROGRAM WILL WAIT FOR APPROXIMATELY 5 MINUTES, TO GIVE ALL THE DRIVES TIME TO COME BACK UP TO SPEED, BEFORE RESTARTING THE TEST SEQUENCE.

NOTE

THIS PROGRAM WILL DESTROY ALL DATA ON THE DISKS. TURN OFF ALL DRIVES THAT YOU DO NOT WANT TO TEST.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD COMPUTER WITH A MINIMUM OF 8K OF MEMORY, AND AN RH11 CONTROLLER WITH AN RS03, RS03/LA OR AN RS04 DISK.

2.2 PRELIMINARY PROGRAMS

DZRSB

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

DESCRIPTION

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5. (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

1. STARTING ADDRESS 200.

A. SET SWITCHES (SEE SEC 5.). ALL DOWN FOR WORST CASE (IF SWITCH-LESS CPV SIMPLY PRESS START)

B. THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

C. PRESS START.

THE PROGRAM WILL NOW MAP THE DATA BUFFERS IN 4K SEGMENTS ON -A- AND -B- PORTS UP TO 124K. IT WILL THEN TYPE OUT THE PARAMETERS OF THE DATA BUFFERS. THE PROGRAM WILL ONLY DO THIS THE FIRST TIME IT IS STARTED, FOR IT STORES THESE ADDRESSES AND CONTINUES USING THEM. TO HAVE THE PROGRAM REMAP THE SYSTEM, THE PROGRAM MUST BE RELOADED.

THE PROGRAM WILL MAKE DATA BUFFERS IN 4K SEGMENTS NOT TO EXCEED 24K.
THE OPERATOR CAN SPECIFY THE BEGINNING BANK AND THE SIZE OF THE
DATA BUFFER AREA (UP TO 24K) BY ENTERING THE CONVERSATION MODE.
OR, BY DEFAULT, THE DATA BUFFERS WILL BEGIN AT THE FIRST AVAILABLE
BANK AND USE AVAILABLE MEMORY UP TO 24K SIZE.

DESCRIPTION

5. OPERATIONAL SWITCH SETTINGS

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

(I.E) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

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

```
                        AC-9318G-MC CZRSCG RH11-RS03/LA-RS04  DATA RELIABILITY DIAG    PAGE 6
                        DESCRIPTION


                        SWITCH SETTINGS ARE:

                                SW<15> = 1 ..... HALT ON ERROR
                                SW<14> = 1 ..... LOOP ON FUNCTION
                                SW<13> = 1 ..... INHIBIT PRINTOUT
                                SW<12> = 1 ..... INHIBIT COMPARISON
                                                 WITH THIS SWITCH SET, THE
                                                 PROGRAM WILL NOT COMPARE THE
                                                 DATA IT READ FROM THE DISK WITH
                                                 THE KNOWN GOOD DATA.
                                SW 11> = 1 ..... HALT ON COMPLETION OF TRANSFER
                                SW<10  = 1 ..... ENTER CONVERSATION MODE
                                SW 09> = 1 ..... LOOP ON ERROR
                                SW<08  = 1 ..... DATA RELIABILITY TEST MODE
                                S. 07> = 1 ..... WAIT IN WAIT MODE
                                                 PROGRAM RUNS IN A BACKGROUND TEST
                                                 WHILE WAITING FOR INTERRUPT, WITH
                                                 SW SET PROGRAM WAITS IN A WAIT
                                                 INSTRUCTION.
                                SW<06> = 1 ..... OPTIONAL TYPEOUT OF RETRY ERRORS
                                SW<05  = 1 ..... INHIBIT PASS COUNT
                                SW<04> = 1 ..... ALLOWS 8 ERROR TYPEOUTS IN THE
                                                 COMPARE ROUTINE BEFORE EXECUTING NEXT READ
                                                 COMMAND. WHEN SWITCH IS 0, ONLY 1 ERROR
                                                 TYPEOUT IS RECORDED.
                                SW<03> = 1 ..... TYPEOUT #  OF ERRORS
                                SW<02> = 1 ..... INHIBIT MEMORY MANAGEMENT
                                SW<01> = 1 ..... DATA TEST ONLY
                                SW<00  = 1 ..... DROPS DRIVE AFTER 20 ERRORS
```

DESCRIPTION

5.1 DATA RELIABILITY TEST MODE

WITH SW8 SET, THE PROGRAM WILL SET THE "BAI" BIT IN RHCS2 AND TRANSFER 64K OF DATA AT A TIME FOR ALL PATTERNS EXCEPT RANDOM. RANDOM WILL BE EXECUTED AS USUAL WITH STANDARD BUFFERS. NO COMPARES ARE DONE IN THIS MODE OF OPERATION EXCEPT ON RANDOM PATTERNS. THIS OPTION SHOULD ONLY BE USED IN DATA TEST OR CONVERSATION MODE. WHEN USED IN CONVERSATION MODE IT OVER RIDES THE NON STANDARD WORD COUNT. YOU SHOULD NOT SELECT A DESIRED DISK ADDRESS IN CONVERSATION MODE FOR IT CAN PRODUCE A DISK ADDRESS OVERFLOW ERROR FOR THIS DATA RELIABILITY TEST MODE ONLY DOES 64K WORD TRANSFERS. IF SW8 IS CHANGED, WHILE THE PROGRAM IS RUNNING. THE PROGRAM WILL FINISH ITS PASS BEFORE EXECUTING THE SW. CH CHANGE.

5.2 CONVERSATION MODE FOR PROGRAM PARAMETERS FOR DATA TEST ONLY

IN CONVERSATION MODE THE OPERATOR CAN SPECIFY ANY ONE OR ALL OF THE PROGRAM PARAMETERS.

NOTE

ONCE IN CONVERSATION MODE, THE ONLY WAY TO REMAP THE SYSTEM IS TO RELOAD THE PROGRAM. TO RESTART THE PROGRAM IN CONVERSATION MODE WITHOUT HAVING TO REANSWER THE QUESTIONS, THE STARTING ADDRESS IS 210. RESET SWITCH 10. TO RESTART THE PROGRAM WITHOUT HAVING TO REANSWER THE PORT SIZING QUESTIONS. RESTART AT 220. RESET SWITCH 10.

AC-9318G-MC CZRSCG RH11-RS03/LA-RS04 DATA RELIABILITY DIAG PAGE 8
DESCRIPTION

THE PROGRAM WILL NOW ASK SEVERAL QUESTIONS, THE TABLE BELOW WILL HELP YOU ANSWER THE QUESTIONS.

TYPE TO	START AT	TYPE TO	START AT
0	000000		
1	020000	20	400000
2	040000	21	420000
3	060000	22	440000
4	100000	23	460000
5	120000	24	500000
6	140000	25	520000
7	160000	26	540000
10	200000	27	560000
11	220000	30	600000
12	240000	31	620000
13	260000	32	640000
14	300000	33	660000
15	320000	34	700000
16	340000	35	720000
17	360000	36	740000

NOTE: TYPE ONLY NUMBERS SHOWN!!!

1. -A- PORT? (Y OR N)

   THIS GIVES YOU THE OPTION TO TEST -A- OR -B- PORT USING THE DATA TEST.

   IF THE ANSWER TO THIS QUESTION IS YES, THE FOLLOWING QUESTION WILL BE ASKED. IF THE ANSWER IS NO, -B- PORT WILL BE TESTED AND QUESTION 3 WILL BE ASKED.

2. 1ST 4K BANK #

   THIS NUMBER THAT IS TYPED WILL DETERMINE WHERE THE BUFFER AREA WILL START ON -A- PORT. USE TABLE ABOVE

NOTE:

PROGRAM IS LOCATED IN 1ST 4K BANK. THEREFORE, THIS BANK CAN NOT BE USED AS A BUFFER.

DESCRIPTION

3. 1ST 4K BANK #

NOTE

THIS DIAGNOSTIC WILL ONLY TEST -B- PORT IF THE PROCESSOR HAS ACCESS TO THAT MEMORY ON -B- PORT. THIS MEMORY MUST HAVE THE SAME ADDRESS ON ALL PORTS.

THIS NUMBER WILL DETERMINE WHERE THE DATA BUFFER AREA WILL START ON -B- PORT.

DESCRIPTION

PROGRAM CONVERSATION

MULTI DRIVE MODE? (YES-NO)

MULTI DISK MODE IS A MODE IN THE PROGRAM WHICH ALLOWS THE OPERATOR TO EXERCISE ALL THE DISKS ON THE SYSTEM WITHOUT RE-STARTING THE PROGRAM. THE PROGRAM, AFTER EXERCISING ONE DISK WILL REPORT A MESSAGE TELLING THE OPERATOR WHICH DISK WILL BE SELECTED NEXT, AND THEN THE PROGRAM WILL EXERCISE THAT DISK. WHEN A COMPLETE PASS IS ACCOMPLISHED, A PASS COMPLETE WILL BE REPORTED AND THE TEST WILL RECYCLE.

IF THE ANSWER TO THE MULTI DRIVE MODE WAS "NO", THE FOLLOWING QUESTION IS ASKED.

UNIT #

THE OPERATOR CAN NOW SELECT THE UNIT HE WISHES TO TEST BY TYPING THE UNIT NUMBER.

OPTIONAL WORD COUNT (YES-NO)

IF THE OPERATOR ANSWERS "NO" TO THIS QUESTION THE NEXT QUESTION WILL BE DELETED FROM THE CONVERSATION.
IF THE WORD COUNT IS NOT SPECIFIED, THE PROGRAM WILL USE AVAILABLE MEMORY UP TO 24K TO CREATE THE DATA BUFFERS.

WD CT

THE OPERATOR CAN SPECIFY ANY LENGTH TRANSFER FROM 1(8) TO 60000(8) WORDS. THE NORMAL TRANSFER LENGTH IS N(8) WORDS WHERE N IS THE MAXIMUM BUFFER SIZE FOR THE AVAILABLE CORE. IN EITHER CASE, BUFFER WILL NOT EXCEED 24 K.

THIS PROGRAM MAPS THE SYSTEM IN 4K SEGMENTS. IF THERE IS A 1K BLOCK OF MEMORY ON THE SYSTEM THAT YOU WOULD LIKE TO REACH, YOU CAN TYPE IN THAT 4K BANK # AND THEN SPECIFY A WC OF 2000.

IF THE WORD COUNT NUMBER TYPED, IS LARGER THAN THE CORE SIZE GIVEN IN THE SETUP ROUTINE, THE QUESTION WILL BE REPEATED.

OPTIONAL DSK ADDR (YES-NO)

IF THE ANSWER TO THIS QUESTION IS NO, THE WHOLE DISK WILL BE WRITTEN AND THE NEXT QUESTION IS NOT ASKED.

DSK ADDR

THE OPERATOR CAN NOW SPECIFY THE STARTING SECTOR

AC-9318G-MC CZRSCG RH11-RS03/LA-RS04 DATA RELIABILITY DIAG PAGE 11
DESCRIPTION

PATTERN NO.?

THIS GIVES THE OPERATOR THE OPTION OF SELECTING ALL THE DATA PATTERNS (#22) OR ANY ONE DATA PATTERN, SIMPLY BY TYPING THE DATA PATTERN NUMBER DESIRED.

PATTERN	0	= 000000
"	1	= 177777
"	2	= 031463
"	3	= 066666
"	4	= 100001
"	5	= 107070
"	6	= 070707
"	7	= 052525
"	10	= 125252
"	11	= 177737
"	12	= 146314
"	13	= 136363
"	14	= 063636
"	15	= 000001
"	16	= 100005
"	17	= 155555
"	20	= 133333
"	21	= RANDOM DATA
"	22	= RUN ALL DATA PATTERNS UNDER PROGRAM CONTROL

IN THIS SECTION OF THE PROGRAM PARAMETER CONVERSATION MODE, THE OPERATOR CAN SELECT ANY ONE OR ALL THREE OF THE CONTROL FUNCTIONS TO BE EXECUTED. THE NORMAL SEQUENCE OF DISK FUNCTIONS UNDER PROGRAM CONTROL ARE WRITE, WRITE CHECK, AND THEN READ. BY ENTERING THE CONVERSATION MODE THE OPERATOR HAS GAINED COMPLETE CONTROL OVER THE DISK FUNCTIONS. HE MUST SPECIFY YES OR NO TO ALL OF THE FOLLOWING QUESTIONS.

WRITE? (YES - NO)
READ? (YES - NO)
WRITE CHECK? (YES - NO)

TO PERFORM A WRITE CHECK ONLY, THE OPERATOR MUST FIRST WRITE SOME KNOWN DATA ON THE DISK. THIS COURSE OF ACTION ALSO PREVAILS FOR A READ ONLY OPERATION.
* IF AN ERROR OCCURS IN THE LINE THE OPERATOR IS TYPING, DEPRESS THE RUB-OUT KEY AND RETYPE ANSWER.
ALL ANSWERS SHOULD BE FOLLOWED BY A CARRIAGE-RETURN

DESCRIPTION

5.3 ROUTINE ABSTRACTS

ADDRESS TEST

THIS TEST WRITES EACH SECTOR WITH ITS OWN ADDRESS THEN READS IT BACK AND COMPARES IT FOR THE CORRECT DATA.

RANEX - RANDOM DATA, ADDRESS AND WORD COUNT TEST

THIS ROUTINE TESTS THE ABILITY OF THE SYSTEM TO ACCESS RANDOM ADDRESSES WITH RANDOM DATA. ONE SECTOR OF RANDOM DATA IS WRITTEN AT A STARTING RANDOM ADDRESS ON THE DISK. IT IS THEN WRITE CHECKED AND READ. ALL ERRORS ARE REPORTED. THIS IS REPEATED 1000 TIMES.

DATA RELIABILITY - DATA PATTERN TEST

IN THIS PORTION OF THE TEST, THE RELIABILITY OF THE DISK SURFACE IS TESTED BY WRITE, WRITE CHECK, AND READ FUNCTIONS. THE ROUTINE FIRST WRITES THE COMPLETE SURFACE WITH A SET DATA PATTERN, THEN A WRITE CHECK OF THE COMPLETE SURFACE IS ACCOMPLISHED, THUS REPORTING ALL ERRORS BETWEEN THE DATA WRITTEN AND THE DATA IN MEMORY. THE DISK IS THEN READ. THE DATA READ FROM THE DISK IS COMPARED AGAINST THE KNOWN DATA PATTERN. THIS COMPARE IS TAKING PLACE THE SAME TIME THE DISK IS BEING READ. THE BUFFER IS CLEARED AS IT IS BEING COMPARED. IF THERE ARE DATA BUFFERS ON -A- AND -B- PORTS, THE DATA TEST WILL TRANSFER DATA OVER -A- PORT ON ODD PASSES AND OVER -B- PORT ON EVEN PASSES.

5.4 SUBROUTINE ABSTRACTS

5.4.1 SCOPE

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

5.4.2 HLT

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

5.4.3 TRAPCATCHER

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. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

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

WHERE:

```
CS1,CS2,ER ETC.        = RS11 DISK REGISTERS.
GOOD                   = EXPECTED DATA.
BAD                    = DATA RECEIVED.
```

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

IF SW0 IS SET, A DRIVE WILL BE DROPPED FROM THE TEST SEQUENCE AFTER 20 ERRORS. THE PROGRAM WILL STATE WHICH DRIVE WAS DROPPED AND ON WHICH PASS IT WAS DROPPED. IF ALL THE DRIVES HAVE BEEN DROPPED, THE PROGRAM WILL TYPE "TESTING UNIT 0" AND HALT", INDICATING THAT IT COULD NOT FIND ANY MORE DRIVES ON THE SYSTEM TO TEST.

7. RESTRICTIONS

THIS DIAGNOSTIC WILL TEST -B- PORT, ONLY IF THE CPU CAN ACCESS THAT MEMORY ON -B- PORT.

8. MISCELLANEOUS

8.1 EXECUTION TIME

PASS COMPLETE WILL BE TYPED OUT AT END OF PASS. IT WILL TAKE 10 TO 20 MINUTES TO COMPLETE A PASS DEPENDING ON THE TYPE OF DRIVE BEING TESTED AND THE SIZE OF THE SYSTEM.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500

```
                                       AC-9318G-MC CZRSCG RH11-RS03 LA-RS04  DATA RELIABILITY DIAG   PAGE 14
                                       DESCRIPTION


                                       8.3  POWER FAIL

                                       THE STARTING ADDRESS FOR THE WRITE  POWER  FAIL  TEST  IS  244.   WHEN
                                       ASKED, ENTER UNIT #.  THE PROGRAM WILL TELL THE OPERATOR WHEN TO POWER
                                       DOWN.  WHEN THE SYSTEM IS POWERED UP, ONLY ONE ERROR IS ALLOWED.   THE
                                       STARTING  ADDRESS  FOR THE WRITECHECK POWER FAIL TEST IS 250.  HERE AS
                                       IN THE WRITE POWER FAIL TEST, THE PROGRAM WILL TELL THE OPERATOR  WHEN
                                       TO POWER DOWN.  WHEN THE POWER COMES BACK, NO ERRORS SHOULD OCCUR.
                                             @
                                       .TITLE   CZRSCG RH11-RS03-RS03/LA-RS04 DATA AND RELIABILITY TEST
                                       ;COPYRIGHT 1973,1974,1975,1976,1977,1978 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                                       ;PROGRAM BY STANLEY HARACKIEWICZ

                                       ;          SWITCH                         USE
                                       ;          ------                 --------------------
              100000                          SW15=    100000            ;HALT ON ERROR
              040000                          SW14=    40000             ;LOOP ON FUNCTION
              020000                          SW13=    20000             ;INHIBIT ERROR TYPEOUTS
              010000                          SW12=    10000             ;INHIBIT COMPARISON
              004000                          SW11=    4000              ;HALT ON COMPLETION OF TRANSFER
              002000                          SW10=    2000              ;CONVERSATION MODE
              001000                          SW9=     1000              ;LOOP ON ERROR
              000400                          SW8=     400               ;DATA RELIABILITY TEST MODE
              000200                          SW7=     200               ;WAIT IN BACKGROUND TEST
              000100                          SW6=     100               ;OPTIONAL TYPEOUT OF RETRY ERRORS
              000040                          SW5=     40                ;INHIBIT PASS COUNT AND UNIT #
              000020                          SW4=     20                ;ALLOWS 8 LOCATIONS TO BE TESTED IN COMPARE ROUTINE
              000010                          SW3=     10                ;TYPE OUT TOTAL # OF ERRORS
              000004                          SW2=     4                 ;INHIBIT MEMORY MANAGEMENT
              000002                          SW1=     2                 ;DATA TEST ONLY
              000001                          SW0=     1                 ;DROP DRIVE AFTER 20 ERRORS
              000000                   .=             0                  ;TRAP CATCHER FROM 0 - 776
              000046                   .=             46                 ;HOOKS FOR ACT 11
      000046  014110                   $ENDAD
              000052                   .=             52
      000052  040000                   BIT14

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


                                       ;NOTE:   FOR PROGRAM TO BE RESTARTED AT 200 IT MUST BE RELOADED
                                       ;        INTO MEMORY DUE TO ONCE ONLY CODE.


              000200                   .=             200
      000200  000137  001230                          JMP      SETSWI             ;START TEST

              000210                   .=             210
      000210  012706  000500                          MOV      #500,SP            ;SETUP STACK
      000214  000137  003232                          JMP      @#ADTST            ;RESTART ADDR
```

```
                    000220                  .=      220
000220  012706  000500                      MOV     #500,SP         ;CONVERSATION MODE WITHOUT
000224  000137  002406                      JMP     @#A1            ;DATA BUFFER QUESTIONS

000230  000137  015174                      JMP     @#RLDR          ;RESTORE LOADER

000234  000137  003324                      JMP     @#ADTL          ;TRACK AND SECTOR SELECT TEST
                                                                    ;WRITE EACH WORD ADDR ON ITSELF AND READ IT BACK
000240  000137  005054                      JMP     @#RANEL         ;RANDOM ADDRESS, DATA TEST
000244  000137  012506                      JMP     @#PFT1          ;DISK WRITE POWER FAIL TEST
000250  000137  013034                      JMP     @#PFT2          ;DISK WRITE CHECK POWER FAIL TEST
```

```
                                        ;RH11 DATA PATTERNS
                                        
000254  000000                          PAT0:   0
000256  177777                          PAT1:   177777
000260  031463                          PAT2:   031463
000262  066666                          PAT3:   066666
000264  100001                          PAT4:   100001
000266  107070                          PAT5:   107070
000270  070707                          PAT6:   070707
000272  052525                          PAT7:   052525
000274  125252                          PAT10:  125252
000276  177737                          PAT11:  177737
000300  146314                          PAT12:  146314
000302  136363                          PAT13:  136363
000304  063636                          PAT14:  063636
000306  000001                          PAT15:  000001
000310  100005                          PAT16:  100005
000312  155555                          PAT17:  155555
000314  133333                          PAT20:  133333
                                        ;PAT21  RANDOM DATA
                                        
                                        ;CLEAR ALL REGISTERS
000316  012777  000040  000510          .CLRDV: MOV     #40,@RSCS2      ;CLEAR ALL REG
000324  013777  001160  000502                  MOV     UNNUM,@RSCS2    ;GET UNIT #
000332  000002                                  RTI
```

```
                                .SBTTL          $KMMR - KERNAL MEMORY MANAGEMENT REGISTER ASSIGNMENTS

            177572                      SR0=177572              ;ADDRESS OF MEM MGMT REGISTER SR0
            177574                      SR1=177574              ;    "    "    "        "    "   SR1
            177576                      SR2=177576              ;    "    "    "        "    "   SR2
            172516                      SR3=172516              ;ADDRESS OF MEM MGMT REGISTER SR3

            172300                      KIPDR0=172300           ;ADDRESS OF KERNEL 'I' PAGE
            172302                      KIPDR1=172302           ;DESCRIPTOR REGISTERS
            172304                      KIPDR2=172304
            172306                      KIPDR3=172306
            172310                      KIPDR4=172310
            172312                      KIPDR5=172312
            172314                      KIPDR6=172314
            172316                      KIPDR7=172316

            172320                      KDPDR0=172320           ;ADDRESSES OF KERNEL 'D' PAGE
            172322                      KDPDR1=172322           ;DESCRIPTOR REGISTERS
            172324                      KDPDR2=172324
            172326                      KDPDR3=172326
            172330                      KDPDR4=172330
            172332                      KDPDR5=172332
            172334                      KDPDR6=172334
            172336                      KDPDR7=172336

            172340                      KIPAR0=172340           ;ADDRESSES OF KERNEL 'I' PAGE
            172342                      KIPAR1=172342           ;ADRESS REGISTERS
            172344                      KIPAR2=172344
            172346                      KIPAR3=172346
            172350                      KIPAR4=172350
            172352                      KIPAR5=172352
            172354                      KIPAR6=172354
            172356                      KIPAR7=172356

            172360                      KDPAR0=172360           ;ADDRESSES OF KERNEL 'D' PAGE
            172362                      KDPAR1=172362           ;ADDRESS REGISTERS
            172364                      KDPAR2=172364
            172366                      KDPAR3=172366
            172370                      KDPAR4=172370
            172372                      KDPAR5=172372
            172374                      KDPAR6=172374
            172376                      KDPAR7=172376
```

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

```
                    000510                          .=      510
000510  005015  047125  052111  LOADSW: .ASCIZ  <15><12>"UNIT # "
000516  021440  000040
000522  005015  040506  040524  UNRECO: .ASCIZ  <15><12>"FATAL"<15><12><12>
000530  006514  005012     000
                                
                                
000535     015  047412  052120  .OPDR:  .ASCIZ  <15><12>"OPT DSK ADDR"
000542  042040  045523  040440
000550  042104  000122
000554  005015  040504  040524  DATA:   .ASCIZ  <15><12>"DATA "
000562  000040
                                
000564  051127  020124  051105  WRTERR: .ASCIZ  "WRT ERR"
000572  000122
                                
000574  051127  020124  045503  WCKERR: .ASCIZ  "WRT CK ERR"
000602  042440  051122     000
                                
000607     122  020104  051105  RDERR:  .ASCIZ  "RD ERR"
000614  000122
                                
000616  005015  042522  020103  RECOV:  .ASCIZ  <15><12>"REC RETRY CT "
000624  042522  051124  020131
000632  052103  000040
                                
000636  005015     000          CRLFLF: .ASCIZ  <15><12>
                                
000641     040  054450  047440  YORN:   .ASCIZ  " (Y OR N)"
000646  020122  024516     000
                                
000653     015  030412  052123  STABUF: .ASCIZ  <15><12>"1ST 4K BANK # "
000660  032040  020113  040502
000666  045516  021440  000040
                                
000674  005015  040522  042116  RANDM:  .ASCIZ  <15><12>"RANDOM "
000702  046517  000040
000706  005015  042524  052123  TSTNG:  .ASCIZ  <15><12>"TESTING "
000714  047111  020107     000
000721     015  050012  053517  PDOWN:  .ASCIZ  <15><12>"POWER DOWN"
000726  051105  042040  053517
000734  000116
000736  005015  040520  044522  PAREER: .ASCIZ  <15><12>"PARITY ERR"
000744  054524  042440  051122
000752     000
                                
                    000754                          .EVEN
```

```
                001000                  .=      1000
                                        
001000  000000                  ICNT:   0               ;LH = ITERATION COUNT  ;RH = TEST NO.
001002  000000                  ERRORS: 0               ;ERROR COUNT
001004  000000  000000          PCNT:   0,0             ;2 WORD PASS COUNT
001010  000000                  LAD:    0               ;LOOP ADDRESS FOR SCOPE
001012  000000                  HLTADR: 0               ;ADDRESS OF LAST HLT INSTRUCTION EXECUTED
001014  001000                  FILCHR: 1000            ;FILCHR=0 (CHAR)  ;FILCHR+1=2 (COUNT)
001016  177564                  TPS:    177564          ;OUTPUT STATUS REGISTER
001020  177560                  TKS:    177560
001022  177562                  TKB:    177562
001024  177566                  TPB:    177566          ;OUTPUT BUFFER
001026  177570                  SWR:    177570          ;SWITCH REGISTER
001030  177570                  DISPLAY:177570          ;DISPLAY REGISTER
                                        
                                ;DISK I/O REGISTERS
                                        
001032  172040                  RSCS1:  172040          ;DISK CONTROL + STATUS REGISTER
001034  172050                  RSCS2:  172050          ;DISK CONTROL + STATUS REGISTER
001036  172042                  RSWC:   172042          ;WORD COUNT REGISTER
001040  172044                  RSBA:   172044          ;BUS ADDRESS
001042  172046                  RSDA:   172046          ;DISK ADDRESS (DESIRED ADDRESS)
001044  172052                  RSDS:   172052          ;DRIVE STATUS
001046  172054                  RSER:   172054          ;ERROR REG.
001050  172056                  RSAS:   172056          ;ATTENTION SUMMARY
001052  172060                  RSLA:   172060          ;LOOK AHEAD
001054  172062                  RSDB:   172062          ;DATA BUFFER REGISTER
001056  172064                  RSMR:   172064          ;MAINTENANCE REGISTER
001060  172066                  RSDT:   172066          ;DRIVE TYPE REGISTER
001062  000204                  RSVEC:  204             ;INTERUPT RSVEC
                                        
                                ;BIT ASSIGNMENTS FOR ERROR TYPE OUTS
                                        
                000002                  DB=2            ;DATA BUFFER
                000004                  DA=4            ;DESIRED ADD
                000010                  WC=10           ;WORD COUNT
                000020                  BA=20           ;BUS ADDRESS
                000040                  DS=40           ;DRIVE STATUS
                000100                  AS=100          ;ATTENTION SUMMARY
                000204                  LA=204          ;LOOK AHEAD
                000220                  MR=220          ;MAINTENANCE
                000240                  DT=240          ;DRIVE TYPE
                                        
001064  000206                  STATUS: 206                     ;DISK INTERRUPT STATUS
001066  000200                  PRIORITY:BIT7                   ;DISK PRIORITY LEVEL
```

```
                    000006                  RW=6                    ;R/W IN PDR REG
                    000000                  UP=0                    ;UP BITY IN PDR REG
                    000250                  MMVEC=250               ;ADDR OF MEM MGMT ERPOR TRAP
001070              000000                  STAMEM: 0               ;STARTING LOC FOR -A- PORT
001072              000000                  SAVAST: 0               ;SAVE LOC FOR STAMEM
001074              000000                  STBCOM: 0               ;STARTING LOC FOR -B- PORT
001076              000000                  SAVCPU: 0               ;SAVE LOC FOR CPUBM
001100              000000                  SAVMGA: 0               ;STARTING ADDR FOR -A- PORT  WITH MEM MGMT
001102              000000                  SAVMGB: 0               ;STARTING ADDR FOR B PORT W/MEM MGMT
001104              000000                  SAVMGC: 0               ;STARTING LOC FOR CPU W/MEM MGMT
001106              000000                  SIZEAP: 0               ;SIZE OF A PORT
001110              000000                  SIZEBP: 0               ;SIZE OF B PORT
001112              000000                  WDCTB:  0               ;WC FOR A PORT
001114              000000                  ADB1:   0               ;FLAG FOR PORT BEING TESTED
001116              000000                  VADDR:  0               ;VIRTUAL ADDR
001120              000000                  PHADDR: 0               ;PHYSICAL ADDR
001122              000000                  FLAG2:  0               ;FLAG FOR RESTART AND FOUND DRIVE
001124              000000                  DROP:   0               ;BAD UNITS ON SYSTEM THAT GET DUMPED

                                    ;DISCRIPTION OF FLAG2

                                    ;BIT0 = RESTART
                                    ;BIT1 = FOUND DRIVE
                                    ;BIT2 = ERROR DO A CRLF FOR UNIT #
                                    ;BIT3 = DOING COMPARE
                                    ;BIT4 = SET A16 IN CS1
                                    ;BIT5 = SET A17 IN CS1
                                    ;BIT6 = SET IF MEMORY HAS ALREADY BEEN FOUND
                                    ;BIT7 = WHEN SET MAKE WC UP TO 28K
                                    ;BIT8 = FOUND MEMORY ON -B- PORT
                                    ;BIT9 = POWER DID FAIL
                                    ;BIT10 = WAITING IN BACKGROUND TEST
                                    ;BIT11 = PARITY ERROR ROUTINE
                                    ;BIT12 = POWER FAIL TEST
                                    ;BIT13 = IN MAP ROUTINE
                                    ;BIT14 = IN POWER FAIL OR CONVERSATION MODE
                                    ;BIT15 = ERROR IN POWER FAIL

                                    ;DISCRIPTION OF FLAG

                                    ;BIT0 = USED FOR WRITE COUNTER
                                    ;BIT1 = USED FOR WRITE COUNTER
                                    ;BIT2 = TRANSFER MODE 64K
                                    ;BIT5 = OPTIONAL DMA
                                    ;BIT6 = TEST -B- PORT
                                    ;BIT7 = LAST DISK BUFFER FLAG
                                    ;BIT8 = PROGRAM IS IN ADDRESS OR RANDOM TEST
                                    ;BIT9 = ERROR DURING TRANSFER
                                    ;BIT10 = DATA TEST ONLY
                                    ;BIT11 = MULTIPORT
                                    ;BIT12 = READ
                                    ;BIT13 = WRITE CHECK
                                    ;BIT14 = WRITE
                                    ;BIT15 = PROGRAM CONTROL MODE
```

```
                                        ;RH11 DEDICATE REGISTERS (MEMORY)
                                        
001126  000000                          FLAG:    0                      ;TEST REGISTER
001130  000000                          WRDCT:   0                      ;WORKING WORD COUNT
001132  000000                          TRACK:   0                      ;WORKING DAE
001134  000000                          DMA:     0                      ;WORKING DAR
001136  000000                          PATNU:   0                      ;DATA PATTERN INDEX
001140  000000                          BUF:     0                      ;WORKING DATA BUFFER (OUT-IN)
001142  000000                          TDMA:    0                      ;TEMP DAR
001144  000000                          SWRDCT:  0                      ;STANDARD WORD COUNT
001146  000000                          ERCOUNT: 0                      ;ERROR COUNT FOR MESSAGES.
001150  000000                          SAVE:    0
001152  000000                          HRDER:   0                      ;POINTER FOR HARD ERROR
001154  000000                          BLOCK:   0
001156  000000                          PASSC:   0
001160  000000                          UNNUM:   0                      ;UNIT CURRENTLY BEING TES ED
001162  000000                          UNITSV:  0                      ;SET BIT=UNIT ON BUS
001164  000000                          UNCMP:   0                      ;FOR COMPARING FOR # OF DEVICE
001166  000000                          RS04DT:  0                      ;FLAG FOR RS04
001170  000000                          NUM$:    0                      ;WORK LOC FOR NUMBER INPUTS
001172  000000                          CMD:     0                      ;LOC FOR CS2 COMMANDS
001174  000000                          SWITCH:  0                      ;FLAG FOR WHICH RANDOM NUMBER GEN
001176  000000                          INTFLG:  0                      ;FLAG FOR INTERRUPT
001200  000000                          LOPCNT:  0                      ;ERROR FLAG AND LOOP COUNTER FLAG
001202  000000                          WRITER:  0                      ;CONTAINS # OF WRITE ERRORS
001204  000000                          WCERR:   0                      ;CONTAINS # OF WRITE CHECK ERRORS
001206  000000                          READER:  0                      ;CONTAINS # OF READ ERRORS
001210  000000                          COMERR:  0                      ;CONTAINS # OF COMPARE ERRORS
001212  000000                          MMAVA:   0                      ;MEM MGMT AVAILABLE INDICATOR
001214  000000                          SAVWC:   0                      ;SAVE LOC FOR CONVERSATION WC ROUTINE
001216  000000                          FLAG3:   0                      ;LOOP IN ADDRESS + RANDOM TST FLAG
001220  000000                          SAVWCB:  0                      ;SAVE WC SIZE FOR -B- PORT
                                        
                                        ;RH11 WORK REGISTERS
                                        ;(CAN BE CHANGED IN ANY ROUTINE)
001222  000000                          WORK:    0
001224  000000                          WORK1:   0
001226  000000                          WORK2:   0
```

```
  909  001230  005037  020116          SETSWI: CLR     SWI
  910  001234  012706  000500          BEGIN:  MOV     #500,SP         ;SET STACK TO  *** 500 ***
  911  001240  012737  016274  000024          MOV     #.POWER,@#24    ;SET UP PF VECTOR
  912  001246  012737  000340  000026          MOV     #340,@#26       ;LOCK OUT THE WORLD
  913  001254  012737  015736  000030          MOV     #.HLT,@#30      ;SET EMT VECTOR
  914  001262  012737  000340  000032          MOV     #340,@#32       ;LOCK UP
  915  001270  012737  016642  000034          MOV     #.TRAP,@#34     ;SET TRAP VECTOR
  916  001276  012737  000340  000036          MOV     #340,@#36       ;LOCK UP
  917  001304  005037  001000                  CLR     ICNT            ;INIT ICNT
  918  001310  005037  001010                  CLR     LAD             ;INIT LAD
  919  001314  023737  000042  000046          CMP     @#42,   @#46    ;UNDER ACT11 AUTO MODE?
  920  001322  001423                          BEQ     6$              ;YES-SKIP TITLE PRINT-OUT
  921  001324  104402  001330                  TYPE    .+2             ;.ASCIZ <15><12>"CZRSCG RH11-RS03/4 DA RELIAB" :5  :2
  922  001372  032737  000001  020116  6$:     BIT     #BIT0,SWI
  923  001400  001001                          BNE     2$
  924  001402  104444                          SUSWR                   ;SIZE FOR SWITCHLESS
  925  001404  042737  177677  001126  2$:     BIC     #177677,FLAG    ;CLEAR FLAG
  926  001412  042737  177776  001122          BIC     #177776,FLAG2   ;CLEAR ALL EXECPT RESTART
  927  001420  005037  001216                  CLR     FLAG3           ;CLEAR LOOP IN ADDRESS + RANDOM TST FLAG
  928  001424  032737  000001  001122          BIT     #BIT0,FLAG2     ;IS THIS THE FIRST TIME?
  929  001432  001002                          BNE     1$              ;NO
  930  001434  004737  020120                  JSR     PC,LDR          ;SAVE LOADER
  931  001440  000005                  1$:     RESET                   ;CLEAR THE WORLD
  932  001442  012737  000340  177776          MOV     #340,PS         ;LOCK UP INTERRUPT LEVELS
  933  001450  004537  012452                  JSR     R5,ERRCL        ;CLEAR ERROR COUNTER + PASS CNT
  934  001454  005037  001212                  CLR     MMAVA           ;CLEAR MEM MGMT FLAG
  935  001460  005037  001114                  CLR     AD81            ;TEST A PORT FIRST
  936  001464  032777  000004  177334          BIT     #BIT2,@SWR      ;WANT MEM MGMT?
  937  001472  001021                          BNE     3$              ;NO
  938  001474  012737  001522  000004          MOV     #5$,4           ;SET TIMEOUT TRAP
  939  001502  012737  000340  000006          MOV     #340,6          ;SET PS
  940  001510  005037  177572                  CLR     @#SR0           ;IS MEM MGMT AVAILABLE?
  941  001514  005137  001212                  COM     MMAVA           ;YES
  942  001520  000401                          BR      4$              ;CONT
  943  001522  022626                  5$:     CMP     (6)+,(6)+       ;CLEAR STACK
  944  001524  012737  000006  000004  4$:     MOV     #6,4            ;RESET
  945  001532  005037  000006                  CLR     6               ;TRAP
  946  001536  032737  000001  001122  3$:     BIT     #BIT0,FLAG2     ;IS THIS THE FIRST TIME
  947  001544  001002                          BNE     CALM            ;NO
  948  001546  000137  020210                  JMP     SIZZAP          ;SIZE BUFFERS
  949  001552  004737  011476          CALM:   JSR     PC,@#EXTMEM     ;SET UP DATA BUFFERS
  950  001556  004737  015222          CALM1:  JSR     PC,.MAMK        ;TURN ON PARITY MEM
  951  001562  032737  000001  001122          BIT     #BIT0,FLAG2     ;1ST TIME ?
  952  001570  001006                          BNE     3$              ;NO
  953  001572  013737  001144  001214          MOV     SWRDCT,SAVWC    ;SAVE WC FOR CONVERSATION MODE COMPARE
  954  001600  013737  001112  001220          MOV     WDCTB,$AVWCB    ;SAVE WC FOR -B- PORT
  955  001606  052737  000001  001122  3$:     BIS     #BIT0,FLAG2     ;SET 1ST TIME FLAG
  956  001614  005037  001134                  CLR     DMA             ;CLEAR DAR REGISTERS
  957  001620  005037  001136                  CLR     PATNU           ;CLEAR PATTER COUNT
  958  001624  013737  001144  001130          MOV     SWRDCT,WRDCT
  959  001632  032777  000002  177166          BIT     #BIT1,@SWR      ;DATA TEST ONLY?
  960  001640  001403                          BEQ     2$              ;NO
  961  001642  052737  002000  001126          BIS     #BIT10,FLAG     ;YES
  962  001650  032777  002000  177150  2$:     BIT     #BIT10,@SWR     ;ENTER CONVERSATION MODE?
  963  001656  001007                          BNE     1$              ;YES GO TO CONVERSATION MODE
  964  001660  052737  074000  001126          BIS     #74000,FLAG
```

```
   965  001666  004537  010220                  JSR     R5,RESTOR       ;RESTORE ORIGINAL WD CNT
   966  001672  000137  003232                  JMP     ADTST
   967  001676  000137  002276          1$:     JMP     @#CONM          ;ENTER CONVERSATION MODE
```

```
                                        ;FIND OUT HOW MANY DRIVES
                                        ;FIRST TEST RSAS
                                
001702  012701  000010          DRVENO: MOV     #8.,R1          ;PUT 8 INTO R1 FOR COUNT
001706  042737  000002  001122          BIC     #BIT1,FLAG2     ;CLEAR FOUND DRIVE FLG
001714  012777  000000  177112          MOV     #0,@RSCS2       ;SET DEVICE TO ZERO
001722  012777  000007  177116  TRY:    MOV     #7,@RSER        ;CAUSE AN ERROR +SETS BIT IN AS REG
001730  005301                          DEC     R1              ;DO A MAXIMUN OF 16 TIMES
001732  001403                          BEQ     DVNUM           ;TESTED FOR ALL DRIVES GET OUT
001734  005277  177074                  INC     @RSCS2          ;INCREMENT DRIVE UNIT
001740  000770                          BR      TRY             ;REPEAT FOR NEXT DRIVE
001742  017737  177102  001162  DVNUM:  MOV     @RSAS,UNITSV    ;SAVE
001750  043737  001124  001162          BIC     DROP,UNITSV     ;DROP BAD DRIVES
001756  012737  000401  001164          MOV     #401,UNCMP      ;SETUP TO CMP WITH UNITSV
001764  012737  000000  001160          MOV     #0,UNNUM        ;PUT 0 INTO UNIT NO.
001772  032777  000040  177026          BIT     #BIT5,@SWR      ;INHIBIT TYPE OUT?
002000  001005                          BNE     STTEST          ;YES
002002  104402  000706                  TYPE    ,TSTNG
002006  042737  000004  001122          BIC     #BIT2,FLAG2     ;CLEAR ERROR FLAG
002014  033737  001164  001162  STTEST: BIT     UNCMP,UNITSV    ;IS THIS DRIVE ON THE SYSTEM
002022  001473                          BEQ     TRYNX           ;NO
002024  013777  001160  177002  UNTYP:  MOV     UNNUM,@RSCS2    ;YES PUT UNIT # INTO CS2
002032  005037  001166                  CLR     RS04DT          ;CLEAR DRIVE TYPE FLAG
002036  022777  000004  177014          CMP     #4,@RSDT        ;RS03LA?
002044  001004                          BNE     8$              ;NO
002046  012737  000004  001166          MOV     #4,RS04DT       ;SET DRIVE TYPE FLAG
002054  000422                          BR      1$              ;CONT
002056  005777  176776          8$:     TST     @RSDT           ;IS THIS A RS03?
002062  001417                          BEQ     1$              ;YES
002064  022777  000001  176766  2$:     CMP     #1,@RSDT        ;IS THIS A RS03 4US?
002072  001413                          BEQ     1$              ;YES
002074  022777  000002  176756  3$:     CMP     #2,@RSDT        ;IS THIS A RS04?
002102  001404                          BEQ     6$              ;YES
002104  022777  000003  176746          CMP     #3,@RSDT        ;RS04?
002112  001037                          BNE     TRYNX           ;GET A NEW NUMBER
002114  052737  177777  001166  6$:     BIS     #-1,RS04DT      ;YES RS04
002122  032737  040000  001122  1$:     BIT     #BIT14,FLAG2    ;IN POWER FAIL OR CONVERSATION?
002130  001401                          BEQ     7$              ;NO
002132  000207                          RTS     PC              ;YES
002134  032777  000200  176702  7$:     BIT     #BIT7,@RSDS     ;IS THIS DRIVE READY ?
002142  001423                          BEQ     TRYNX           ;NO GET ANOTHER DRIVE
002144  032777  000040  176654          BIT     #BIT5,@SWR      ;TYPEOUT?
002152  001016                          BNE     4$              ;NO
002154  032737  000004  001122          BIT     #BIT2,FLAG2     ;WAS THERE AN ERRER?
002162  001402                          BEQ     5$              ;NO
002164  104402  000636                  TYPE    ,CRLFLF
002170                          5$:
002170  013746  001160                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
002174  104406                          TYFES                   ;TYPE STACK IN OCTAL - SUPRESS
002176  104402  000040                  TYPE    ,40             ;TYPE SPACE
002202  042737  000004  001122          BIC     #BIT2,FLAG2     ;CLEAR ERROR FLAG
002210  000426                  4$:     BR      NOWGO           ;NOW TEST
```

```
1020  002212  006337  001164          TRYNX:  ASL     UNCMP           ;CHECK NEXT BIT FOR DRIVE
1021  002216  103403                          BCS     CHCKDV          ;DID WE TEST ANY REG?
1022  002220  005237  001160                  INC     UNNUM           ;INC UNIT #
1023  002224  000673                          BR      STTEST          ;CHECK FOR NEXT DRIVE
1024
1025                                  ;THIS PROGRAM WILL DEFAULT TO TESTING UNIT 0 IF IT CAN NOT FIND ANY DRIVES
1026  002226  032737  000002  001122  CHCKDV: BIT     #BIT1,FLAG2     ;FOUND DRIVE?
1027  002234  001012                          BNE     DONEE           ;YES WE DID TEST A DRIVE
1028  002236  012737  100000  001164          MOV     #100000,UNCMP   ;NO DRIVES TESTED, COULD NOT SET
1029  002244  005037  001160                  CLR     UNNUM           ;ANY AS BITS, THUS DEFAULTS TO 0
1030  002250  013746  001160                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
1031  002254  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
1032  002256  000000                          HALT                    ;COULD NOT SET ANY ATA BITS
1033                                                                  ;BY SETTING ERROR BITS
1034                                                                  ;GO BACK AND USE OTHER DIAG.
1035  002260  000402                          BR      NOWGO           ;DEFAULT TO DRIVE 0
1036  002262  000137  013466          DONEE:  JMP     OUT             ;GET OUT
1037  002266  052737  000002  001122  NOWGO:  BIS     #BIT1,FLAG2     ;FOUND DRIVE
1038  002274  000207                          RTS     PC
1039
1040                                  ;ENTER OPERATOR CONVERSATION MODE
1041
1042  002276                          CONM:
1043  002276  104402  002302                  TYPE    .+2             ;.ASCIZ <15><12>"-A- PORT"
1044  002316  004737  003306                  JSR     PC,CMPY         ;COMPARE FOR YES
1045  002322  001405                          BEQ     2$              ;YES
1046  002324  012737  177777  001114          MOV     #-1,AOB1        ;B PORT
1047  002332  000137  002354                  JMP     1$              ;TEST -B- PORT
1048  002336  104402  000653          2$:     TYPE    ,STABUF
1049  002342  104420                          RDOCT
1050  002344  012637  001070                  MOV     (6)+,STAMEM     ;START BUFFER AT 4K
1051  002350  000137  002374                  JMP     NOPORT          ;GET OUT
```

```
 1052  002354  104402  000653            1$:     TYPE    .STABUF
 1053  002360  104420                            RDOCT
 1054  002362  012637  001074                    MOV     (6)+,STECOM     ;GET ANS
 1055  002366  052737  000100  001126            BIS     #BIT6,FLAG      ;AND SAVE IT
 1056  002374  004737  011476            NUPORT: JSR     PC,EXTMEM       ;SET B PORT FLAG
 1057  002400  013737  001144  001130            MOV     SWRDCT,WRDCT    ;CAL BUFFERS AND WC
 1058  002406  052737  002000  001126    A1:     BIS     #BIT10,FLAG     ;GET STANDARD WC
 1059  002414  004537  012452                    JSR     R5,ERRCL        ;SET BIT FOR DATA TEST ONLY
 1060  002420  042737  174040  001126            BIC     #174040,FLAG    ;CLEAR ERROR CNT + PASS CNT
 1061  002426  104402  002432                    TYPE    .+2             ;CLEAR MULTI FLAG MODE +PATTERN SELECT
 1062  002450  004737  003306                    JSR     PC,CMPY         ;.ASCIZ <15><12>"MULTI DRIVE"
 1063  002454  001004                            BNE     DATTES          ;COMPARE FOR YES
 1064  002456  052737  004000  001126            BIS     #BIT11,FLAG     ;ANS IS NO
 1065  002464  000434                    1$:     BR      ASKWC           ;SET BIT FOR MULTI DRIVE
 1066  002466  104402  000510            DATTES: TYPE    .LOADSW
 1067  002472  104420                            RDOCT
 1068  002474  012637  001170                    MOV     (6)+,NUM$       ;GET NUMBER
 1069  002500  022737  000010  001170            CMP     #10,NUM$        ;CORRECT # ?
 1070  002506  103767                            BLO     DATTES          ;NO
 1071  002510  013737  001170  001160            MOV     NUM$,UNNUM      ;SET UNIT #
 1072  002516  004737  006544                    JSR     PC,FNDTYP               ;TEST FOR RS04 OR 03
 1073  002522  005002                    1$:     CLR     R2              ;CLEAR WORK AREA
 1074  002524  000261                            SEC                     ;SET CARRY
 1075  002526  006102                    2$:     ROL     R2              ;SET BIT IN WORK
 1076  002530  005737  001170                    TST     NUM$            ;IS THIS THE RIGHT BIT FOR THE RIGHT DISK
 1077  002534  001403                            BEQ     3$              ;YES
 1078  002536  005337  001170                    DEC     NUM$            ;NO TRY AGAIN
 1079  002542  000771                            BR      2$              ;TEST AGAIN
 1080  002544  010237  001162            3$:     MOV     R2,UNITSV       ;SET DRIVE BIT IN UNITSV
 1081  002550  052737  000002  001122            BIS     #BIT1,FLAG2     ;SET FOUND DRIVE FLAG
 1082
 1083  002556                            ASKWC:
 1084  002556  104402  002562                    TYPE    .+2             ;.ASCIZ <15><12>"OPT WD CT"
 1085  002576  004737  003306                    JSR     PC,CMPY         ;COMPARE FOR YES
 1086  002602  001401                            BEQ     WCCON           ;YES
 1087  002604  000444                            BR      OPDAR           ;CONT
```

```
1088  002606                                WCCON:
1089  002606  104402  002612                      TYPE    ..+2            ;.ASCIZ <15><12>"WD CT "
1090  002624  104420                              RDOCT
1091  002626  012637  001170                      MOV     (6)+,NUM$       ;GET NUMBER
1092  002632  005737  001170                      TST     NUM$            ;IS IT 0?
1093  002636  001763                              BEQ     WCCON           ;YES ASK AGAIN FOR LENGTH
1094  002640  013702  001214                      MOV     SAVWC,R2        ;GET STANDARD WC FOR -A- PORT
1095  002644  005737  001114                      TST     A0B1            ;-A- PORT?
1096  002650  001402                              BEQ     1$              ;YES
1097  002652  013702  001220                      MOV     SAVWCB,R2       ;GET WC FOR -B- PORT
1098  002656  005202                      1$:     INC     R2
1099  002660  020237  001170                      CMP     R2,NUM$         ;IS NUM$ LESS THAN SWRDCT
1100  002664  101750                              BLOS    WCCON           ;YES ASK FOR COUNT AGAIN
1101  002666  013737  001170  001144              MOV     NUM$,SWRDCT     ;OPERATING WORD COUNT
1102  002674  013737  001144  001130              MOV     SWRDCT,WRDCT
1103  002702  005737  001114                      TST     A0B1            ;B PORT?
1104  002706  001403                              BEQ     OPDAR           ;NO
1105  002710  013737  001130  001112              MOV     WRDCT,WDCTB     ;YES GET WC
1106
1107  002716  104402  000535              OPDAR:  TYPE    ,.OPDR
1108  002722  004737  003306                      JSR     PC,CMPY         ;COMPARE FOR YES
1109  002726  001037                              BNE     OPPAT           ;ANS IS NO
1110  002730  052737  000040  001126              BIS     #BIT5,FLAG      ;SET OPTIONAL DMA FLAG
1111  002736  104402  002742                      TYPE    ,..+2           ;.ASCIZ "="
1112  002744  104420                              RDOCT
1113  002746  012637  001170                      MOV     (6)+,NUM$       ;GET NUMBER
1114  002752  022737  000004  001166              CMP     #4,RS04DT       ;RS03LA?
1115  002760  001004                              BNE     3$              ;NO
1116  002762  022737  003777  001170              CMP     #3777,NUM$      ;IS ADDR. CORRECT?
1117  002770  000412                              BR      2$
1118  002772  005737  001166              3$:     TST     RS04DT          ;IS THIS A RS04?
1119  002776  001404                              BEQ     1$              ;NO
1120  003000  022737  017777  001170              CMP     #17777,NUM$     ;IS ADD. CORRECT?
1121  003006  000403                              BR      2$              ;GET OUT
1122  003010  022737  007777  001170      1$:     CMP     #7777,NUM$      ;IS ADD. CORRECT?
1123  003016  101737                      2$:     BLOS    OPDAR           ;NO
1124  003020  013737  001170  001142              MOV     NUM$,TDMA       ;TEMP SECTOR REGISTER
```

```
                                OPPAT:
                                        TYPE    ,.+2            ;.ASCIZ <15><12>"PATN # "
                                        RDOCT
                                        MOV     (6)+,NUMS       ;GET NUMBER
                                        CMP     #23,NUMS        ;TEST FOR CORRECT NO
                                        BLOS    OPPAT           ;ASK AGAIN
                                        CLR     PATNU           ;CLEAR PATTERN #
                                        CMP     #22,NUMS
                                        BEQ     OPWRT           ;DATA PATTERN UNDER PROGRAM CONTROL
                                        BIS     #BIT15,FLAG     ;SET PROGRAM FLAG
                                        MOV     NUMS,PATNU      ;OPERATOR WANTS TO SELECT DATA
                                        CLC
                                        ROL     PATNU

                                OPWRT:
                                        TYPE    .+2             ;.ASCIZ <15><12>"WRITE"
                                        JSR     PC,CMPY         ;COMPARE FOR YES
                                        BNE     OPRD            ;ASK ABOUT WRITE CHECK
                                        BIS     #BIT14,FLAG     ;YES SET FLAG BIT

                                OPRD:
                                        TYPE    .+2             ;.ASCIZ <15><12>"READ"
                                        JSR     PC,CMPY         ;COMPARE FOR YES
                                        BNE     OPWCK
                                        BIS     #BIT12,FLAG     ;SET FLAG TO READ

                                OPWCK:
                                        TYPE    .+2             ;.ASCIZ <15><12>"WRT CK"
                                        JSR     PC,CMPY         ;COMPARE FOR YES
                                        BNE     ADTST
                                        BIS     #BIT13,FLAG

                                ADTST:  BIT     #BIT11,FLAG     ;ARE WE IN MULTI DRIVE MODE
                                        BEQ     EXMFLG          ;BRANCH IF NO.
                                        JSR     PC,DRVENO       ;GET DRIVES TO BE TESTED
                                EXMFLG: BIC     #BIT2,FLAG      ;CLEAR XFER MODE FLAG
                                        BIT     #BIT8,@SWR      ;XFER MODE?
                                        BEQ     1$              ;NO
                                        BIS     #BIT2,FLAG      ;SET XFER MODE FLAG
                                1$:     BIT     #BIT10,FLAG     ;TEST FOR DATA TEST ONLY
                                        BEQ     ADT1            ;DO COMPLETE TEST
                                        JMP     DATAT           ;DO DATA TEST ONLY

                                CMPY:   TYPE    ,YORN
                                        RDLIN
                                        CMPB    #'Y,INPUT       ;TEST FOR YES
                                        RTS     PC
                                ADTL:   BIS     #BIT15,FLAG3    ;SET LOOP IN ADDRESS TEST FLAG GOT HERE
                                                                ;BECAUSE PROGRAM WAS STARTED AT 234
```

```
                                        ;RH11 ADDRESS TEST #1 (TRACK AND SECTOR SELECTION TEST)
                                        ;WRITE 100(OCTAL) RS03, 200(OCTAL) RS04, WORDS IN EACH SECTOR
                                        ;THE WORD CONTAINS THE ADDRESS OF EACH SECTOR
                                        ;WHEN THE COMPLETE DISK IS WRITTEN READ
                                        ;BACK EACH SECTOR AND COMPARE FOR THE CORRECT
                                        ;DATA IN THE SECTOR
                                        ;PS IS AT LEVEL 7 SO NO INTERRUPTS

003332                                  ADT1:                           ;ADDRESS TEST
                                        ;*****************************************************************
                                        ;TEST 1         ADDRESS TEST
                                        ;*****************************************************************
003332  104400                          TST1:   SCOPE
003334  032737  000004  001126          ADT1A:  BIT     #BIT2,FLAG      ;XFER MODE?
003342  001402                                  BEQ     3$              ;NO
003344  000137  003716                          JMP     DATAT           ;YES
003350  012737  000340  177776          3$:     MOV     #340,PS         ;LOCK UP PS
003356  012737  020000  017436                  MOV     #20000,OUTBUF   ;START BUF AT 20000
003364  052737  000400  001126                  BIS     #BIT8,FLAG      ;SET TEST FLAG
003372  013737  001144  001150                  MOV     SWRDCT,SAVE     ;SAVE STD WD COUNT
003400  005037  001134                          CLR     DMA             ;CLEAR DISK ADD
003404  104426                                  CLRDV                   ;INIT DRIVE
003406  004737  006566                          JSR     PC,WHTHU        ;GET WORD COUNT
003412  013737  001130  001144  5$:     MOV     WRDCT,SWRDCT
003420  013737  017436  001140  2$:     MOV     OUTBUF,BUF      ;SET UP CURRENT ADDRESS
003426  104414                          SEABUF: ERCLR                   ;CLEAR RS REGISTERS IF ERROR
003430  013700  017436                          MOV     OUTBUF,R0       ;SET UP ADDRESS BUFFER
003434  013701  001130                          MOV     WRDCT,R1        ;
003440  013720  001134                  XSEABUF:MOV     DMA,(0)+        ;LOAD OUTBUF WITH DATA TO BE WRITTEN
003444  005301                                  DEC     R1              ;FILL OUTBUF
003446  001374                                  BNE     XSEABUF         ;WITH DATA
003450  012737  000061  001172                  MOV     #61,CMD         ;WRITE NO I/E
003456  104416                                  DKCMD                   ;GO WRITE
003460  105777  175346                          TSTB    @RSCS1          ;CHECK FOR READY
003464  100375                                  BPL     .-4
003466  005777  175340                          TST     @RSCS1          ;TEST FOR ERROR
003472  100010                                  BPL     WRNEXB          ;BRANCH IF NO ERROR
003474  012737  003426  001010                  MOV     #SEABUF,LAD     ;SET UP LOOP ADDRESS
003502  052737  001000  001126                  BIS     #BIT9,FLAG      ;SET ERROR BIT IN FLAG
003510  104430                                  LOGW                    ;LOG WRITE ERROR
003512  104034                                  HLT     !WC!DA!BA
003514  104400                  WRNEXB: SCOPE
003516  004737  007054                          JSR     PC,DISBUF       ;SET UP NEXT DISK ADDR.
003522  000741                                  BR      SEABUF          ;WRITE NEXT SECTOR
003524  104400                  RRDSEC: SCOPE
```

```
1219  003526  104414                    RDSECT: ERCLR                   ;CLEAR ERRORS
1220  003530  012737  000071  001172            MOV     #71,CMD         ;READ NO I/E
1221  003536  104416                            DKCMD                   ;DO A READ
1222  003540  105777  175266                    TSTB    @RSCS1          ;CHECK FOR READY
1223  003544  100375                            BPL     .-4             ;NOT READY BRANCH BACK
1224  003546  005777  175260                    TST     @RSCS1          ;TEST FOR ERROR
1225  003552  100006                            BPL     ADHGT           ;BRANCH IF NO ERROR
1226  003554  052737  001000  001126            BIS     #BIT9,FLAG      ;SET ERROR FLAG
1227  003562  104432                            LOGR                    ;LOG READ ERROR
1228  003564  104014                            HLT     !WC!DA
1229  003566  104400                            SCOPE
1230  003570  013702  017436            ADHGT:  MOV     OUTBUF,R2
1231  003574  013746  001130                    MOV     WRDCT,-(SP)     ;SAVE OLD WC
1232  003600  004737  006566                    JSR     PC,WHTHU
1233  003604  013703  001130                    MOV     WRDCT,R3
1234  003610  012637  001130                    MOV     (SP)+,WRDCT     ;RESTORE OLD WC
1235  003614  023712  001134            SANHT:  CMP     DMA,(2)         ;CMP FOR CORRECT ADDR.
1236  003620  001004                            BNE     ADERR           ;BRANCH IF DATA DID NOT COMPARE
1237  003622  005722                            TST     (2)+            ;GET NEXT ADDRESS OF INBUF
1238  003624  005303                            DEC     R3              ;DEC SECTOR COUNT
1239  003626  001372                            BNE     SANHT           ;TEST NEXT WORD
1240  003630  000412                            BR      CHKADT
1241  003632  013701  001134            ADERR:  MOV     DMA,GOOD        ;CORRECT ADDRESS
1242  003636  011200                            MOV     (2),BAD         ;DATA IN ERROR
1243  003640  104000                            HLT                     ;DISK ADD DID NOT MATCH WRITTEN ADDRESS
1244  003642  104436                            LOGC                    ;LOG COMPARE ERROR
1245  003644  004737  014224                    JSR     PC,PRNT         ;INHIBIT TYPEOUT?
1246  003650  001002                            BNE     CHKADT          ;YES
1247  003652  104402  000636                    TYPE    ,CRLFLF
1248
1249                                    ;*****REPORT ONLY ONE ERROR PER SECTOR*****
1250
1251  003656  104400                    CHKADT: SCOPE
1252  003660  004737  007054                    JSR     PC,DISBUF       ;SET UP NEXT DISK BUFFER
1253  003664  000717                            BR      RRDSEC          ;CHECK NEXT SECTOR
1254  003666  013737  001150  001144            MOV     SAVE,SWRDCT     ;GET STD WD COUNT
1255  003674  042737  000400  001126            BIC     #BIT8,FLAG      ;CLEAR TEST FLAG
1256  003702  032737  100000  001216            BIT     #BIT15,FLAG3    ;DOES OPERATOR WANT TO LOOP ON TEST
1257  003710  001402                            BEQ     .+6             ;NO
1258  003712  000137  003334                    JMP     ADT1A           ;YES
```

```
1259  003716                                DATAT:                          ;DATA TEST
1260                                        ;*****************************************************************
1261                                        ;TEST 2          DATA TEST
1262                                        ;*****************************************************************
1263  003716  104400                        TST2:   SCOPE
1264  003720  005037  001134                       CLR     DMA             ;CLEAR DISK ADDRESS
1265  003724  104426                               CLRDV                   ;CLEAR RS REGISTERS
1266  003726  013737  001144  001130               MOV     SWRDCT,WRDCT
1267  003734  032737  000100  001126               BIT     #BIT6,FLAG      ;MULTI PORT?
1268  003742  001403                               BEQ     3$              ;NO
1269  003744  005737  001114                       TST     A0B1            ;A OR B PORT?
1270  003750  001003                               BNE     1$              ;B PORT
1271  003752  004737  011354               3$:     JSR     PC,APORT        ;A PORT
1272  003756  000402                               BR      2$
1273  003760  004737  011422               1$:     JSR     PC,BPORT        ;B PORT
1274  003764  004737  007000               2$:     JSR     PC,VECTRR       ;SETUP INT VECTOR
1275  003770  012777  000340  175066               MOV     #340,@STATUS    ;SET DISK STATUS REG LOC  2C6
1276  003776  012737  004030  001152               MOV     #LDAT,HRDER     ;SETUP FOR HARD ERROR RETURN
1277  004004  013737  017436  001140               MOV     OUTBUF,BUF      ;SETUP OUTPUT BUFFER
1278  004012  052737  000003  001126               BIS     #3,FLAG         ;SET COUNTER TO 3
1279  004020  004537  007722                       JSR     R5,PASEL        ;SET UP DATA BUFFERS
1280  004024  005037  001200               LDAT1:  CLR     LOPCNT          ;CLEAR ERROR FLAG
1281  004030  104414                       LDAT:   ERCLR                   ;CLEAR RS REG. IF ERROR
1282  004032  004537  006504                       JSR     R5,OPDSEL       ;SET UP DISK ADDRESS
1283  004036  032737  040000  001126               BIT     #BIT14,FLAG     ;TEST FOR WRITE
1284  004044  001456                               BEQ     SLH             ;TEST FOR READ
1285  004046  012737  000161  001172               MOV     #161,CMD        ;WRITE WITH I/E
1286  004054  104416                               DKCMD                   ;DO A WRITE
1287  004056  004737  011770                       JSR     PC,WATT         ;WAIT FOR INTERRUPT
1288  004062  012737  004030  001010               MOV     #LDAT,LAD       ;SETUP SCOPE LOOP
1289  004070  032737  001000  001126               BIT     #BIT9,FLAG      ;WAS THERE AN ERROR?
1290  004076  001423                               BEQ     WRXBL           ;CONT
1291  004100  104430                               LOGW                    ;LOG WRITE ERROR
1292  004102  005237  001200                       INC     LOPCNT          ;SET ERROR FLAG
1293  004106  004737  014224               2$:     JSR     PC,PRNT         ;TYPE ?
1294  004112  001004                               BNE     1$              ;NO
1295  004114  104402  000554                       TYPE    ,DATA
1296  004120  104402  000564                       TYPE    ,WRTERR
1297  004124  104044                       1$:     HLT     !DS!DA          ;WRITE ERROR
1298  004126  005337  001126                       DEC     FLAG            ;DEC COUNTER
1299  004132  032737  000003  001126               BIT     #3,FLAG         ;DONE YET WITH 3RD TRY?
1300  004140  001333                               BNE     LDAT            ;NOT 3 TRIES YET? TRY AGAIN
1301  004142  004737  011460                       JSR     PC,WTNO         ;TYPE CAN NOT WRITE
```

```
1302  004146  005737  001200          WRXBL:  TST     LOPCNT          ;WAS THERE AN ERROR?
1303  004152  001402                          BEQ     WRX1            ;NO
1304  004154  004737  012012                  JSR     PC,TYPREC       ;TYPE RECOVERED
1305  004160  005037  0012 0          WRX1:   CLR     LOPCNT          ;CLEAR ERROR FLAG
1306  004164  104400                          SCOPE
1307  004166  052737  000003  001126          BIS     #3,FLAG         ;CLEAR RETRY COUNT
1308  004174  004737  007054                  JSR     PC,LTSBUF       ;SET BUFFER FOR WRITE CHECK
1309  004200  000711                          BR      LDAT1
1310  004202  104400                  SLH:    SCOPE
1311  004204  104414                  SLH2:   ERCLR                   ;CLEAR RS REG IF ERRORS
1312  004206  004537  006504                  JSR     R5,OPDSEL       ;IS THE OPERATOR SELECTING THE PRIN
1313  004212  032737  020000  001126          BIT     #BIT13,FLAG     ;TEST FOR WRITE CHECK
1314  004220  001002                          BNE     1$              ;YES
1315  004222  000137  004526                  JMP     ESH1            ;NO
1316  004226  013737  017436  001140  1$:     MOV     OUTBUF,BUF      ;SET UP CURRENT ADDRESS
1317  004234  012737  000151  001172          MOV     #151,CMD        ;WRITE CHECKWITH I/E
1318  004242  104416                          DKCMD                   ;GO WRITE CHECK
1319  004244  004737  011770                  JSR     PC,WATT         ;WAIT FOR INTERRUPT
1320  004250  032737  001000  001126  XESH:   BIT     #BIT9,FLAG      ;IS THERE AN ERROR?
1321  004256  001505                          BEQ     1$              ;NO ERROR
1322  004260  005737  001200                  TST     LOPCNT          ;1ST ERROR?
1323  004264  001001                          BNE     2$              ;NO
1324  004266  104434                          LOGWC                   ;YES LOG ERROR
1325  004270  032777  000100  174530  2$:     BIT     #BIT6,@SWR      ;TYPE ALL ERRORS?
1326  004276  001007                          BNE     3$              ;YES
1327  004300  032777  001000  174520          BIT     #BIT9,@SWR      ;LOOP ON ERROR?
1328  004306  001003                          BNE     3$              ;YES
1329  004310  005737  001200                  TST     LOPCNT          ;FIRST ERROR?
1330  004314  001056                          BNE     10$             ;NO
1331  004316  004737  014224          3$:     JSR     PC,PRNT         ;TYPE OUT?
1332  004322  001052                          BNE     4$              ;NO
1333  004324  104402  000554                  TYPE    ,DATA
1334  004330  104402  000574                  TYPE    ,WCKERR
1335  004334  017702  174500                  MOV     @RSBA,R2        ;GET CORRECT BA
1336  004340  023702  017436                  CMP     OUTBUF,R2       ;DID A WD GET XFERED?
1337  004344  001406                          BEQ     9$              ;NO
1338  004346  032777  000400  174452          BIT     #BIT8,@SWR      ;XFER MODE?
1339  004354  001002                          BNE     9$              ;YES
1340  004356  162702  000002                  SUB     #2,R2
1341  004362  004737  014224          9$:     JSR     PC,PRNT         ;TYPEOUT ERRORS?
1342  004366  001030                          BNE     4$              ;NO
1343  004370  005737  001212                  TST     MMAVA           ;IS MEM MGMT AVAILABLE?
1344  004374  001402                          BEQ     7$              ;NO
```

```
1345  004376  004737  006632            JSR     PC,PHYCO        ;YES GET VITURAL ADDR
1346  004402  010237  001222    7$:     MOV     R2,WORK         ;GET BA
1347  004406                    8$:
1348  004406  104402  004412            TYPE    ,.+2            ;.ASCIZ <15><12>"(BA ="
1349  004422                    6$:
1350  004422  017746  174574            MOV     @WORK,-(6)      ;PUT @WORK ON STACK
1351  004426  104404                    TYPEO                   ;TYPE STACK IN OCTAL
1352  004430  104402  004434            TYPE    ,.+2            ;.ASCIZ " WC="
1353  004442  017746  174370            MOV     @RSWC,-(6)      ;PUT @RSWC ON STACK
1354  004446  104404                    TYPEO                   ;TYPE STACK IN OCTAL
1355  004450  104026            4$:     HLT     !DA!DB!BA       ;NOTE: BA REG. = +2 OF ACTUAL MEMORY
1356                                                            ;LOC AFTER WORDS HAVE BEEN XFERED
1357  004452  005237  001200    10$:    INC     LOPCNT          ;INC ERROR COUNT
1358  004456  022737  000010  001200    CMP     #10,LOPCNT      ;10 TRYS YET?
1359  004464  001247                    BNE     SLH2            ;NO
1360  004466  004737  006524            JSR     PC,NOREC        ;TYPE UNRECOVERABLE
1361  004472  005737  001200    1$:     TST     LOPCNT          ;ANY ERRORS?
1362  004476  001402                    BEQ     5$              ;NO
1363  004500  004737  012012            JSR     PC,TYPREC       ;TYPE RECOVERED
1364  004504  005037  001200    5$:     CLR     LOPCNT          ;CLEAR ERROR COUNTER
1365  004510  104400                    SCOPE
1366  004512  012737  004204  001C1C    MOV     #SLH2,LAD       ;SETUP LOOP ADDRESS
1367  004520  004737  007054            JSR     PC,DISBUF       ;SET UP THE DISK BUFFER
1368  004524  000422                    BR      SLH2A
1369  004526  004537  011306    ESH1:   JSR     R5,CLEAR        ;CLEAR BUFFER
1370  004532  004537  006504    ESH:    JSR     R5,OPDSEL       ;OPERATOR SELECTED DISK ADDRESS?
1371  004536  032737  010000  001126    BIT     #BIT12,FLAG     ;TEST FOR READ
1372  004544  001002                    BNE     1$              ;YES
1373  004546  000137  004760            JMP     MSTR            ;NO READ
1374  004552  104400            1$:     SCOPE
1375  004554  042737  000003  001126    BIC     #3,FLAG         ;CLEAR RE-READ COUNT
1376  004562  005037  001200            CLR     LOPCNT          ;CLEAR FLAG
1377  004566  000137  004576            JMP     DSKRD           ;CONT
1378  004572  000137  004204    SLH2A:  JMP     SLH2
1379  004576  104414    DSKRD:  ERCLR                           ;CLEAR RS REG IF ERRORS
1380  004600  012737  000171  001172    MOV     #171,CMD        ;READ WITH I/E
1381  004606  104416                    DKCMD                   ;READ
1382  004610  004737  011770            JSR     PC,     WATT    ;WAIT FOR INTERRUPT
1383  004614  032777  010000  174204    BIT     #10000,@SWR     ;COMPARE?
1384  004622  001006                    BNE     ELH             ;NO
1385  004624  032737  000004  001126    BIT     #BIT2,FLAG      ;COMPARE?
1386  004632  001002                    BNE     ELH             ;NO
1387  004634  004537  010452            JSR     R5,COMPARE      ;COMPARE
```

```
1388  004640  032737  001000  001126  ELH:    BIT     #BIT9,FLAG      ;IS THERE AN ERROR?
1389  004646  001433                          BEQ     ADRD            ;NO
1390  004650  032777  000100  174150          BIT     #BIT6,@SWR      ;SOFT ERROR TYPEOUT?
1391  004656  001003                          BNE     3$              ;YES
1392  004660  005737  001200                  TST     LOPCNT          ;FIRST ERROR?
1393  004664  001011                          BNE     2$              ;NO
1394  004666  004737  014224          3$:     JSR     PC,PRNT         ;TYPEOUT?
1395  004672  001004                          BNE     1$              ;NO
1396  004674  104402  000554                  TYPE    ,DATA
1397  004700  104402  000607                  TYPE    ,RDERR
1398  004704  104432                  1$:     LOGR                    ;LOG READ ERROR
1399  004706  104044                          HLT     !DS!DA
1400  004710  104400                  2$:     SCOPE
1401  004712  005237  001200                  INC     LOPCNT          ;COUNT ERRORS
1402  004716  022737  000010  001200          CMP     #10,LOPCNT      ;LAST RETRY?
1403  004724  001324                          BNE     DSKRD           ;NO
1404  004726  004737  006524                  JSR     PC,NOREC        ;TYPE UNRECOVERABLE
1405  004732  000137  004752          4$:     JMP     ADR1            ;CONT
1406  004736  104400                  ADRD:   SCOPE
1407  004740  005737  001200                  TST     LOPCNT          ;ANY ERRORS?
1408  004744  001402                          BEQ     ADR1            ;NO
1409  004746  004737  012012                  JSR     PC,TYPREC       ;TYPE RECOVERED
1410  004752  004737  007054          ADR1:   JSR     PC,DISBUF       ;GO SET UP DISK BUFFER.
1411  004756  000434                          BR      READCT          ;CONT. READING
1412  004760  005737  001126          MSTR:   TST     FLAG
1413  004764  100423                          BMI     3$              ;OPERATOR SELECTED PATTERN
1414  004766  062737  000002  001136          ADD     #2,PATNU        ;INC PATTERN INDEX
1415  004774  022737  000044  001136          CMP     #44,PATNU
1416  005002  001402                          BEQ     .+6
1417  005004  000137  003716                  JMP     DATAT           ;NOT LAST PATTERN EXIT
1418  005010  005037  001136                  CLR     PATNU           ;LAST PATTERN EXIT
1419  005014  032777  000002  174004          BIT     #BIT1,@SWR      ;DATA TEST ONLY?
1420  005022  001006                          BNE     2$              ;YES
1421  005024  032737  002000  001126          BIT     #BIT10,FLAG     ;DATA TEST ONLY?
1422  005032  001404                          BEQ     1$              ;NO
1423  005034  005137  001114          3$:     COM     AOB1            ;ALTERNATE PORTS
1424  005040  000137  006072          2$:     JMP     EXTPPR          ;LOOP
1425  005044  000137  005062          1$:     JMP     RANEX           ;DO NEXT TEST
1426
1427  005050  000137  004532          READCT: JMP     ESH             ;CONT. READING
1428  005054  052737  100000  001216  RANEL:  BIS     #BIT15,FLAG3    ;SET LOOP IN RANDOM TEST GOT
1429                                                                  ;HERE BY STARTING AT LOC 240
```

```
 1430  005062                          RANEX:                          ;RANDOM ADDRESS DATA TEST
 1431                                  ;THIS PROGRAM WRITES, WRITECHECKS AND READS 1 SECTOR OF RANDOM DATA FROM RANDOM DISK
 1432                                  ;ADDRESSES. THIS TEST WILL MAKE 1000(10) PASSES BEFORE IT IS COMPLETED
 1433
 1434                                  ;*****************************************************************************
 1435                                  ;TEST 3         RANDOM ADDRESS RANDOM DATA TEST
 1436                                  ;*****************************************************************************
 1437  005062  104400                  $ST3:   SCOPE
 1438  005064  032737  000004  001126          BIT     #BIT2,FLAG      ;FAST XFER MODE?
 1439  005072  001402                          BEQ     2$              ;NO
 1440  005074  000137  006072                  JMP     EXTPPR          ;GET OUT
 1441  005100  052737  000400  001126  2$:     BIS     #BIT8,FLAG      ;SET TEST FLAG
 1442  005106  012737  020000  017436          MOV     #20000,OUTBUF   ;GET STARTING ADDR OF BUF
 1443  005114  013737  017436  001116          MOV     OUTBUF,VADDR    ;SAVE BUFFER ADDR
 1444  005122  005737  001212                  TST     MMAVA           ;MEM MGMT AVAILABLE?
 1445  005126  001402                          BEQ     1$              ;NO
 1446  005130  005037  177572                  CLR     @#SR0           ;TURN IT OFF
 1447  005134  012737  000042  001136  1$:     MOV     #42,PATNU       ;DO RANDOM COM. RE
 1448  005142  104426                          CLRDV                   ;INIT DRIVE
 1449  005144  012737  176030  001156          MOV     #-1000.,PASSC   ;SET UP PASS CO NT
 1450  005152  012737  005712  001152          MOV     #RWRED,HRDER    ;SET UP FOR HARL ERROR
 1451  005160  004737  007000                  JSR     PC,VECTRR       ;SETUP INTERRUPT VECTOR
 1452  005164  012777  000340  173672          MOV     #340,@STATUS
 1453  005172  012737  005274  001010  WRLG1:  MOV     #WRERR,LAD      ;SETUP LOOP ADDRESS
 1454  005200  012737  000001  001222          MOV     #1,WORK         ;SET UP RANDOM GENERATOR WORD
 1455  005206  013701  017436                  MOV     OUTBUF,R1
 1456  005212  004537  010044                  JSR     R5,RANDOM       ;GENERATE RANDOM DATA
 1457  005216  017737  012214  001134          MOV     @OUTBUF,DMA     ;SET UP DISK ADDRESS
 1458  005224  042737  170000  001134          BIC     #170000,DMA
 1459  005232  052737  000003  001126          BIS     #3,FLAG         ;SET COUNTER
 1460  005240  004737  006566                  JSR     PC,WHTHU        ;GET WORD COUNT
 1461  005244  013737  001130  001222  2$:     MOV     WRDCT,WORK      ;GENERATE RANDOM BUFFER
 1462  005252  013701  017436                  MOV     OUTBUF,R1
 1463  005256  004537  010044                  JSR     R5,RANDOM
 1464  005262  013737  017436  001140          MOV     OUTBUF,BUF      ;SET UP OUTPUT BUFFER
 1465  005270  005037  001200                  CLR     LOPCNT          ;CLR ERROR FLAG
```

```
 1466  005274  104414                    WRERR:  ERCLR
 1467  005276  012737  000161  001172            MOV     #161,CMD        ;WRITE WITH I/E
 1468  005304  104416                            DKCMD                   ;WRITE
 1469  005306  004737  011770                    JSR     PC,WATT         ;WAIT FOR INTERRUPT
 1470  005312  032737  001000  001126  2$:       BIT     #BIT9,FLAG      ;WAS THERE AN ERROR?
 1471  005320  001435                            BEQ     WRRCK1          ;NO
 1472  005322  032777  000100  173476            BIT     #BIT6,@SWR      ;TYPE RETRY ?
 1473  005330  001003                            BNE     5$              ;YES
 1474  005332  005737  001200                    TST     LOPCNT          ;FIRST TIME?
 1475  005336  001013                            BNE     6$              ;NO
 1476  005340  104430                    5$:     LOGW                    ;LOG WRITE EPROR
 1477  005342  005237  001200                    INC     LOPCNT          ;SET ERROR FLAG
 1478  005346  004737  014224                    JSR     PC,PRNT         ;TYPEOUT?
 1479  005352  001004                            BNE     3$              ;YES
 1480  005354  104402  000674                    TYPE    ,RANDM
 1481  005360  104402  000564                    TYPE    ,WRTERR
 1482  005364  104044                    3$:     HLT     !DS!DA
 1483  005366  104400                    6$:     SCOPE
 1484  005370  005337  001126                    DEC     FLAG
 1485  005374  032737  000003  001126            BIT     #3,FLAG
 1486  005402  001334                            BNE     WRERR           ;RETRY
 1487  005404  004737  011460                    JSR     PC,WTNO         ;TYPE CAN NOT WRITE
 1488  005410  000137  006014                    JMP     EXRAX           ;GET NEW NUMBER
```

```
1489  005414  005737  001200          WRRCK1: TST     LOPCNT          ;ANY ERRORS?
1490  005420  001402                          BEQ     1$              ;NO
1491  005422  004737  012012                  JSR     PC,TYPREC       ;TYPE RECOVERED
1492  005426  104400                  1$:     SCOPE
1493  005430  005037  001200                  CLR     LOPCNT          ;CLEAR LOOP COUNT
1494  005434  104414                  WRRCK:  ERCLR                   ;CLEAR RS REG IF ERRORS
1495  005436  012737  000151  001172          MOV     #151,CMD        ;WRITE CHECK WITH I/E
1496  005444  104416                          DKCMD                   ;WRITE CHECK
1497  005446  004737  011770                  JSR     PC,WATT         ;WAIT FOR INTERRUPT
1498  005452  032737  001000  001126  4$:     BIT     #BIT9,FLAG      ;ERROR?
1499  005460  001453                          BEQ     1$              ;NO
1500  005462  032777  000100  173336          BIT     #BIT6,@SWR      ;TYPE ALL RETRYS?
1501  005470  001003                          BNE     2$              ;YES
1502  005472  005737  001200                  TST     LOPCNT          ;FIRST ERROR?
1503  005476  001030                          BNE     5$              ;NO
1504  005500  104434                  2$:     LOGWC                   ;LOG WRITE CK
1505  005502  004737  014224                  JSR     PC,PRNT         ;TYPEOUT?
1506  005506  001052                          BNE     6$              ;NO
1507  005510  104402  000674                  TYPE    ,RANDM
1508  005514  104402  000574                  TYPE    ,WCKERR
1509  005520  017737  173314  001222          MOV     @RSBA,WORK      ;GET CORRECT BA
1510  005526  162737  000002  001222          SUB     #2,WORK
1511  005534  104402  005540                  TYPE    ,+2             ;.ASCIZ <15><12>"(BA)="
1512  005550  017746  173446                  MOV     @WORK,-(6)      ;PUT @WORK ON STACK
1513  005554  104404                          TYPEO                   ;TYPE STACK IN OCTAL
1514  005556  104026                          HLT     !DB!DA!BA       ;BA=MEMORY LOC +2 OF ACTUAL WORD
1515  005560  005237  001200          5$:     INC     LOPCNT          ;INC RETRY COUNT
1516  005564  022737  000010  001200          CMP     #10,LOPCNT      ;LAST ONE YET?
1517  005572  001320                          BNE     WRRCK           ;NO
1518  005574  104402  000522                  TYPE    ,UNRECO
1519  005600  005037  001200                  CLR     LOPCNT          ;CLEAR LOPCNT
1520  005604  000137  006014                  JMP     EXRAX           ;GET NEW NUMBER
1521  005610  005737  001200          1$:     TST     LOPCNT          ;ANY ERRORS?
1522  005614  001407                          BEQ     6$              ;NO
1523  005616  104402  000616                  TYPE    ,RECOV
1524  005622  013746  001200                  MOV     LOPCNT,-(6)     ;GET NUMBER
1525  005626  104406                          TYPES                   ;TYPE IT
1526  005630  104402  000636                  TYPE    ,CRLFLF
1527  005634  104400                  6$:     SCOPE
1528  005636  052737  000003  001126          BIS     #3,FLAG         ;SET COUNTER
```

```
 1529  005644  104400                          SCOPE
 1530  005646  005037  001200                  CLR     LOPCNT          ;CLEAR COUNTER
 1531  005652  004537  011306                  JSR     R5,CLEAF        ;CLEAR BUFFER
 1532  005656  104414                  RREAD:  ERCLR                   ;CLEAR R5 REG IF ERRORS
 1533  005660  012737  000171  001172          MOV     #171,CMD        ;READ WITH I/E
 1534  005666  104416                          DKCMD                   ;READ
 1535  005670  004737  011770                  JSR     PC,     WATT    ;WAIT FOR INTERRUPT
 1536  005674  032777  010000  173124          BIT     #BIT12,@SWR     ;COMPARE ?
 1537  005702  001003                          BNE     RWRED           ;NO
 1538  005704  004537  010452                  JSR     R5,COMPARE      ;YES
 1539  005710  000400                          BR      RWRED           ;CONT
 1540  005712  032737  001000  001126  RWRED:  BIT     #BIT9,FLAG      ;IS THERE AN ERROR?
 1541  005720  001435                          BEQ     EXRAX           ;NO
 1542  005722  104432                          LOGR                    ;LOG READ ERR
 1543  005724  032777  000100  173074  1$:     BIT     #BIT6,@SWR      ;TYPE ALL ERRORS?
 1544  005732  001016                          BNE     2$              ;YES
 1545  005734  032777  001000  173064          BIT     #BIT9,@SWR      ;LOOP ON ERROR?
 1546  005742  001012                          BNE     2$              ;YES
 1547  005744  005737  001200                  TST     LOPCNT          ;FIRST ERROR?
 1548  005750  001010                          BNE     3$              ;NO
 1549  005752  004737  014224                  JSR     PC,PRNT         ;TYPEOUT?
 1550  005756  001004                          BNE     2$              ;NO
 1551  005760  104402  000674                  TYPE    ,RANDM
 1552  005764  104402  000607                  TYPE    ,RDERR
 1553  005770  104006                  2$:     HLT     !DB!DA
 1554  005772  104400                  3$:     SCOPE
 1555  005774  005237  001200                  INC     LOPCNT          ;UPDATE COUNTER
 1556  006000  022737  000010  001200          CMP     #10,LOPCNT      ;LAST TRY YET?
 1557  006006  001323                          BNE     RREAD           ;RETRY
 1558  006010  004737  006524                  JSR     PC,NOREC        ;TYPE UNRECOVERABLE
 1559  006014  005737  001200          EXRAX:  TST     LOPCNT          ;ANY ERRORS?
 1560  006020  001402                          BEQ     EXRXX           ;NO
 1561  006022  004737  012012                  JSR     PC,TYPREC       ;TYPE RECOVERED
 1562  006026  104400                  EXRXX:  SCOPE
 1563  006030  005237  001156                  INC     PASSC           ;+1 PASS COUNT
 1564  006034  001402                          BEQ     1$              ;IS TEST DONE?
 1565  006036  000137  005172                  JMP     WRLG1           ;NO
 1566  006042  005037  001136          1$:     CLR     PATNU           ;END OF TEST
 1567  006046  042737  000400  001126          BIC     #BIT8,FLAG      ;CLEAR TEST FLAG
 1568  006054  032737  100000  001216          BIT     #BIT15,FLAG3    ;LOOP ON THIS TEST?
 1569  006062  001402                          BEQ     .+6             ;NO
 1570  006064  000137  005062                  JMP     RANEX           ;YES
```

```
                                        ;CHECK FOR MULTI DISK MODE
                                        ;IF IN MULTI DISK MODE REPORT "END"
                                        ;IF LAST DISK ON SYSTEM HAS BEEN EXERCISED.

                                        ;*****************************************************************
                                        ;TEST 4          TEST FOR MULTI DISK MODE
                                        ;*****************************************************************
006070  104400                  *ST4:   SCOPE
006072  005037  001134          EXTPPR: CLR
006076  104426                          CLRDV                   ;INIT DRIVE
006100  032737  004000  001126          BIT     #dIT11,FLAG     ;ARE WE IN MULTI DISK MODE
006106  001404                          BEQ     EXTPP           ;NO REPORT "END"
006110  004737  002212                  JSR     PC,TRYNX        ;YES TEST FOR ALL DRIVES
006114  000137  003246                  JMP     EXMFLG          ;RESTART TESTING OF DRIVES
006120  004737  002262          EXTPP:  JSR     PC,DONEE        ;GET PASS COUNT

                                        ;THIS ROUTINE CLEARS THE DRIVE
                                        ;REGISTERS IF THERE WAS AN ERROR

006124  032737  001000  001126  .ERCLR: BIT     #BIT9,FLAG      ;ANY ERRORS?
006132  001404                          BEQ     1$              ;NO
006134  104426                          CLRDV                   ;CLEAR ALL ERRORS
006136  042737  001000  001126          BIC     #BIT9,FLAG      ;CLEAR ERROR FLAG
006144  000002                  1$:     RTI                     ;EXIT

                                        ;ENTER DISK HANDLER BY THE TRAP INSTRUCTION
                                        ;ARGUMENT TO TRAP INSTRUCTION IS TWO ORDER
                                        ;BYTE OF THE CONTROL REGISTER.

006146  013777  001134  172666  .DKCMD: MOV     DMA,@RSDA       ;LOAD DISK ADD
006154  005037  001176                  CLR     INTFLG          ;CLEAR INTERRUPT FLAG
006160  032737  020000  001122          BIT     #BIT13,FLAG2    ;IN MAPPING ROUTINE?
006166  001021                          BNE     4$              ;YES
006170  032737  000004  001126          BIT     #BIT2,FLAG      ;MAX DATA TEST?
006176  001415                          BEQ     4$              ;NO
006200  022737  000042  001136          CMP     #42,PATNU       ;RANDOM DATA?
006206  001411                          BEQ     4$              ;YES
006210  052777  000010  172616          BIS     #10,@RSCS2      ;SET BAI
006216  005077  172614                  CLR     @RSWC           ;64K XFER
006222  012777  017436  172610          MOV     #OLDBUF,@RSBA   ;SETUP BA
006230  000435                          BR      2$              ;CONT
006232  013777  001140  172600  4$:     MOV     BUF,@RSBA       ;LOAD (CMA) BUSS ADDRESS
006240  013702  001130                  MOV     WRDCT,R2        ;GET NEGATIVE
006244  005402                          NEG     R2              ;WORD COUNT
006246  010277  172564                  MOV     R2,@RSWC        ;LOAD WC
006252  032737  000400  001126          BIT     #BIT8,FLAG      ;RANDOM TEST?
006260  001033                          BNE     1$              ;YES A PORT ONLY WITH NO MEM MGMT
006262  005737  001212                  TST     MMAVA           ;MEM MGMT AVAILIABLE?
006266  001416                          BEQ     2$              ;NO
006270  032737  000040  001122          BIT     #BIT5,FLAG2     ;SET A17 IN RSCS1
006276  001403                          BEQ     3$              ;NO
006300  052737  001000  001172          BIS     #BIT9,CMD       ;YES
```

```
1623   006306  032737  000020  001122  3$:     BIT     #BIT4,FLAG2     ;SET A16?
1624   006314  001403                          BEQ     2$              ;NO
1625   006316  052737  000400  001172          BIS     #BIT8,CMD       ;YES
1626   006324  032737  000100  001126  2$:     BIT     #BIT6,FLAG      ;MULTI PORT?
1627   006332  001406                          BEQ     1$              ;NO
1628   006334  005737  001114                  TST     A0B1            ;TEST A OR B PORT?
1629   006340  001403                          BEQ     1$              ;A PORT
1630   006342  052737  002000  001172          BIS     #BIT10,CMD      ;B PORT
1631   006350  013777  001172  172454  1$:     MOV     CMD,@RSCS1      ;LOAD FUNCTION REG.
1632   006356  000002                          RTI                     ;RETURN FROM TRAP
1633
1634                                   ;RH11 DISK INTERRUPT HANDLER
1635                                   ;ROUTINE CONTINUES ON ERRORS
1636
1637   006360  042737  001000  001126  DKINT:  BIC     #BIT9,FLAG      ;CLEAR ERROR BIT
1638   006366  005777  172440                  TST     @RSCS1          ;TEST FOR ERROR
1639   006372  100401                          BMI     2$
1640   006374  000425                          BR      INTEXT          ;JUMP IF NO ERRORS
1641   006376  017702  172430          2$:     MOV     @RSCS1,R2       ;GET CONTENTS OF CS1
1642   006402  042702  037777                  BIC     #37777,R2       ;CLEAR ALL BUT SC AND TRE
1643   006406  022702  140000                  CMP     #140000,R2      ;IS SC AND TRE BOTH SET?
1644   006412  001413                          BEQ     TRUERR          ;YES THERE IS SOME KIND OF XFER ERROR
1645   006414  032777  100000  172422          BIT     #100000,@RSDS   ;IS THE ATA BIT SET?
1646   006422  001007                          BNE     TRUERR          ;YES
1647   006424  104140                          HLT     !AS!DS          ;WRONG UNIT INTERRUPTED
1648                                                                   ;IF YOU HAVE JUST POWERED UP A DRIVE OR
1649                                                                   ;ARE RUNNING THE POWER FAIL TEST,
1650                                                                   ;INTERRUPTS WILL OCCUR FROM DRIVES OTHER
1651                                                                   ;THAN THE UNIT UNDER TEST. IF THIS TYPEOUT
1652                                                                   ;SHOWS NO ERRORS IN THE REGISTERS OF THE DRIVE
1653                                                                   ;UNDER TEST, THAT DRIVE IS OK
1654   006426  012777  177777  172414  1$:     MOV     #-1,@RSAS       ;CLEAR ALL ATA BITS
1655   006434  013716  001152                  MOV     HRDER,(SP)      ;GET RETURN ADD.
1656   006440  000002                          RTI                     ;RETRY
1657   006442  052737  001000  001126  TRUERR: BIS     #BIT9,FLAG      ;SET ERROR BIT
1658   006450  032777  004000  172350  INTEXT: BIT     #BIT11,@SWR     ;HALT ON COMPLETION FLAG
1659   006456  001401                          BEQ     .+4
1660   006460  000000                          HALT                    ;YES BIT 11 SET IN SWR HALT
1661   006462  032737  002000  001122          BIT     #BIT10,FLAG2    ;WAIT IN BACKGROUND TEST?
1662   006470  001402                          BEQ     1$              ;NO
1663   006472  012716  012436          2$:     MOV     #NPRRET,(SP)    ;MODIFY RETURN ADD.
1664   006476  010637  001176          1$:     MOV     SP,INTFLG       ;SET INT FLG
1665   006502  000002                          RTI                     ;EXIT
1666
1667                                   ;ROUTINE TO SET UP TRACK # FROM OPTION
1668                                   ;ENTER FROM JSR R5, OPDSEL
1669
1670   006504  032737  000040  001126  OPDSEL: BIT     #BIT5,FLAG      ;OPTIONAL DMA?
1671   006512  001403                          BEQ     1$              ;NO
1672   006514  013737  001142  001134          MOV     TDMA,DMA        ;GET OPT. DMA
1673   006522  000205                  1$:     RTS     R5              ;EXIT
```

```
1674
1675
1676                                        .EVEN
1677
1678  006524  004737  014224          NOREC:  JSR     PC,PRNT         ;TYPEOUT?
1679  006530  001002                          BNE     1$              ;NO
1680  006532  104402  000522                  TYPE    ,UNRECO
1681  006536  005037  001200          1$:     CLR     LOPCNT          ;CLEAR LOOP COUNTER
1682  006542  000207                          RTS     PC
1683
1684  006544  052737  040000  001122  FNDTYP: BIS     #BIT14,FLAG2    ;SET CHECK DRIVE TYPE FLAG
1685  006552  004737  002024                  JSR     PC,UNTYP        ;CHECK DRIVE TYPE FLAG
1686  006556  042737  040000  001122          BIC     #BIT14,FLAG2    ;CLEAR DRIVE TYPE FLAG
1687  006564  000207                          RTS     PC
1688  006566  022737  000004  001166  WHTHU:  CMP     #4,RS04DT       ;RS03LA?
1689  006574  001004                          BNE     1$              ;NO
1690  006576  012737  000040  001130          MOV     #40,WRDCT       ;GET WORD COUNT
1691  006604  000411                          BR      2$
1692  006606  012737  000200  001130  1$:     MOV     #200,WRDCT      ;RS04
1693  006614  005737  001166                  TST     RS04DT          ;RS04?
1694  006620  001003                          BNE     2$              ;YES
1695  006622  012737  000100  001130          MOV     #100,WRDCT      ;NO
1696  006630  000207                  2$:     RTS     PC
1697
1698
```

```
                                        ;ROUTINE TO CALCULATE VITURAL ADDR
                                        
006632  000302                  PHYCOV: SWAB    R2              ;CALCULATE FROM PHYSICAL ADDR
006634  004737  011756                  JSR     PC,RRR2
006640  006002                          ROR     R2
006642  042702  177770                  BIC     #177770,R2      ;GET REG #
006646  032777  000400  172156          BIT     #BIT8,@RSCS1    ;IS A16 SET?
006654  001402                          BEQ     1$              ;NO
006656  052702  000010                  BIS     #BIT3,R2        ;YES
006662  032777  001000  172142  1$:     BIT     #BIT9,@RSCS1    ;IS A17 SET?
006670  001402                          BEQ     2$              ;NO
006672  052702  000020                  BIS     #BIT4,R2        ;YES
006676  013737  001070  001224  2$:     MOV     STAMEM,WORK1    ;GET BANK # FOR -A- PORT
006704  005737  001114                  TST     A0B1            ;ARE WE ON -A- PORT?
006710  001403                          BEQ     3$              ;YES
006712  013737  001074  001224          MOV     STBCOM,WORK1    ;NO -B- PORT
006720  163702  001224          3$:     SUB     WORK1,R2        ;GET STARTING BANK #
006724  062702  000001                  ADD     #1,R2           ;GET OFFSET FOR REG #
006730  000302                          SWAB    R2              ;GET BANK # INTO
006732  006102                          ROL     R2              ;UPPER BITS
006734  006102                          ROL     R2
006736  006102                          ROL     R2
006740  006102                          ROL     R2
006742  006102                          ROL     R2
006744  017737  172070  001224          MOV     @RSBA,WORK1     ;GET OFFSET FOR ADDR IF ANY
006752  162737  000002  001224          SUB     #2,WORK1        ;CORRECT IT
006760  042737  160000  001224          BIC     #160000,WORK1   ;CLEAR JUNK
006766  050237  001224                  BIS     R2,WORK1        ;GET REG NO
006772  013702  001224                  MOV     WORK1,R2
006776  000207                          RTS     PC

007000  012777  000360  172054  VECTRR: MOV     #DKINT,@RSVEC   ;SETUP INTERRUPT VECTORS
007006  013737  001066  177776          MOV     PRIORITY,PS     ;PRIORITY 4
007014  000207                          RTS     PC

                                        ;THIS ROUTINE IS USED FOR DELAYING THE START OF THIS PROGRAM
                                        ;IF POWER FAILED DURING TESTING. THIS WILL GIVE THE DRIVES TIME TO GET UP
                                        ;TO SPEED. THE DELAY WILL BE ABOUT 3-5 MINUTES DEPENDING UPON THE PROCESSOR

007016  012737  000677  001222  TIMUP:  MOV     #677,WORK
007024  012737  177777  001224  1$:     MOV     #177777,WORK1
007032  000240                  2$:     NOP
007034  005337  001224                  DEC     WORK1
007040  001374                          BNE     2$
007042  005337  001222                  DEC     WORK
007046  001366                          BNE     1$
007050  000137  003232                  JMP     ADTST
```

```
                                        ;ROUTINE TO SETUP DISK BUFFERS
                                        ;ADD WORD COUNT TO STARTING DISK ADDRESSES
                                        ;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS

007054  032737  000040  001126  DISBUF: BIT     #BIT5,FLAG      ;DID OPERATOR SELECT PATTERNS
007062  001402                          BEQ     2$              ;NO
007064  000137  007346                  JMP     BUFEXIT         ;YES
007070  022737  000042  001136  2$:     CMP     #42,PATNU       ;RANDOM PATTERN?
007076  001443                          BEQ     1$              ;YES
007100  032737  000004  001126          BIT     #BIT2,FLAG      ;MAX TST?
007106  001437                          BEQ     1$              ;NO
007110  022737  000004  001166          CMP     #4,RS04DT       ;RS03LA?
007116  001010                          BNE     4$              ;NO
007120  022737  004000  001134          CMP     #4000,DMA       ;DONE YET?
007126  001507                          BEQ     BUFEXIT         ;YES
007130  062737  004000  001134          ADD     #4000,DMA       ;UPDATE DMA
007136  000207                          RTS     PC
007140  005737  001166          4$:     TST     RS04DT          ;RS04?
007144  001010                          BNE     3$              ;YES
007146  022737  006000  001134          CMP     #6000,DMA       ;RS03
007154  001474                          BEQ     BUFEXIT         ;DONE GET OUT
007156  062737  002000  001134          ADD     #2000,DMA       ;UPDATE DMA
007164  000207                          RTS     PC              ;EXIT
007166  022737  007000  001134  3$:     CMP     #7000,DMA       ;DONE YET?
007174  001464                          BEQ     BUFEXIT         ;YES
007176  062737  001000  001134          ADD     #1000,DMA       ;UPDATE ADDR
007204  000207                          RTS     PC
007206  004737  007562          1$:     JSR     PC,BLSZ         ;DEFINE BLOCK SIZE
007212  013737  001154  001224          MOV     BLOCK,WORK1
007220  005237  001134          INCSEC: INC     DMA             ;+1 SECTOR COUNT
007224  022737  010000  001134          CMP     #10000,DMA      ;DONE YET?
007232  001445                          BEQ     BUFEXIT         ;YES
007234  005337  001154                  DEC     BLOCK           ;-1 FROM BLOCK COUNT
007240  001401                          BEQ     COMDAR          ;CMP DMA TO RSDA
007242  000766                          BR      INCSEC          ;RECYCLE
007244  032737  001000  001126  COMDAR: BIT     #BIT9,FLAG      ;ANY ERRORS?
007252  001401                          BEQ     1$              ;NO ERRORS DO COMPARE ON RSDA
007254  000207                          RTS     PC              ;ERRORS DO NOT COMPARE RSDA
007256  023777  001134  171556  1$:     CMP     DMA,@RSDA       ;COMPARE RSDA WITH DMA
007264  001425                          BEQ     CMDAE           ;SHOULD BE EQUAL
007266  104432                          LOGR                    ;AFTER TRANSFER RSDA AND DMA SHOULD BE =
                                                                ;IF NOT, RSDA IS NOT CORRECT. DMA CONTAINS
                                                                ;WHAT RSDA SHOULD =
007270  013701  001134                  MOV     DMA,GOOD        ;GET DMA FOR CORRECT ANS IN GOOD
007274  017700  171542                  MOV     @RSDA,BAD       ;GET RSDA INTO BAD
007300  104000                          HLT                     ;RSDA=BAD DMA=GOOD SEE COMMENTS 7 LINES ABOVE
007302  004737  014224                  JSR     PC,PRNT         ;TYPEOUT?
007306  001014                          BNE     CMDAE           ;NO
007310  011637  001222                  MOV     (SP),WORK       ;GET TEST PC FROM WHERE IT CAME
007314  104402  007320                  TYPE    ,.+2            ;.ASCIZ " TST PC="
007332  013746  001222                  MOV     WORK,-(6)       ;PUT WORK ON STACK
007336  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
007340  105737  001126          CMDAE:  TSTB    FLAG            ;LAST DISK BUFFER?
007344  100032                          BPL     BUFINX          ;NO
```

```
 1800  007346  005037  001134          BUFEXIT:CLR     DMA             ;CLEAR ADDRESS BITS LAST DISK BUFFER
 1801  007352  062716  000002                  ADD     #2,(6)          ;INC STOCK POINTER
 1802  007356  042737  000200  001126  AKH:    BIC     #200,FLAG       ;CLEAR LAST DISK BUFFER FLAG
 1803  007364  032737  000400  001126          BIT     #BIT8,FLAG      ;RANDOM TEST OR ADDR TEST?
 1804  007372  001404                          BEQ     1$              ;NO
 1805  007374  013737  001144  001130  2$:     MOV     SWRDCT,WRDCT
 1806  007402  000466                          BR      EXTDR           ;EXIT
 1807  007404  032737  000100  001126  1$:     BIT     #BIT6,FLAG      ;MULTI PORT?
 1808  007412  001770                          BEQ     2$              ;NO
 1809  007414  005737  001114                  TST     ADB1            ;A OR B PORT?
 1810  007420  001765                          BEQ     2$              ;A PORT
 1811  007422  013737  001112  001130          MOV     WDCTB,WRDCT     ;B PORT
 1812  007430  000453                          BR      EXTDR           ;GET OUT
 1813  007432  005037  001226          BUFINX: CLR     WORK2           ;CLEAR WORK2 FOR BLOCK COUNTER
 1814  007436  013702  001134                  MOV     DMA,R2          ;PUT WORKING DISK ADD INTO WORK
 1815  007442  005237  001226          XINCSEC:INC     WORK2           ;INCREMENT BLOCK COUNT
 1816  007446  022702  007777                  CMP     #7777,R2        ;CMP FOR LAST SECTOR
 1817  007452  001405                          BEQ     XINCSUR         ;+1 SURFACE LAST SECTOR BRANCH
 1818  007454  005202                          INC     R2              ;INC DMA
 1819  007456  005337  001224                  DEC     WORK1           ;DEC BLOCK COUNT
 1820  007462  001367                          BNE     XINCSEC         ;FILLED STANDARD BUFFER YET?
 1821  007464  000734                          BR      AKH             ;WILL TAKE STANDARD SIZE WORD COUNT
 1822  007466  013737  001226  001130  XINCSUR:MOV     WORK2,WRDCT     ;SETTING UP BLOCK COUNT
 1823  007474  000241                          CLC                     ;FOR NON STANDARD BUFFER SIZE
 1824  007476  006137  001130                  ROL     WRDCT
 1825  007502  006137  001130                  ROL     WRDCT
 1826  007506  006137  001130                  ROL     WRDCT
 1827  007512  006137  001130                  ROL     WRDCT
 1828  007516  006137  001130                  ROL     WRDCT
 1829  007522  022737  000004  001166          CMP     #4,RS04DT       ;RS03LA?
 1830  007530  001410                          BEQ     1$              ;YES
 1831  007532  000241                          CLC
 1832  007534  006137  001130                  ROL     WRDCT
 1833  007540  005737  001166                  TST     RS04DT          ;RS04?
 1834  007544  001402                          BEQ     1$              ;NO
 1835  007546  006137  001130                  ROL     WRDCT           ;YES
 1836  007552  052737  000200  001126  1$:     BIS     #200,FLAG       ;SET LAST DISK BUFFER FLAG
 1837  007560  000207          EXTDR:  RTS     PC              ;EXIT
```

```
                                        ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
007562  022737  000004  001166  BLSZ:   CMP     #4,RS04DT       ;RS03LA
007570  001004                        BNE     3$              ;NO
007572  012737  000037  001154        MOV     #37,BLOCK       ;YES
007600  000411                        BR      2$              ;CONTINUE
007602  012737  000177  001154  3$:    MOV     #177,BLOCK      ;SETUP FOR RS04
007610  005737  001166                TST     RS04DT          ;RS04?
007614  001003                        BNE     2$              ;YES
007616  012737  000077  001154  1$:    MOV     #77,BLOCK       ;PUT SECTOR SIZE INTO BLOCK
007624  013702  001130        2$:     MOV     WRDCT,R2        ;FETCH WORD COUNT
007630  033702  001154                BIT     BLOCK,R2        ;ARE THEY EQUAL?
007634  001406                        BEQ     RORBLK          ;YES
007636  043702  001154                BIC     BLOCK,R2        ;SET UP BLOCK OVERFLOW
007642  005237  001154                INC     BLOCK
007646  063702  001154                ADD     BLOCK,R2
007652  000241                RORBLK: CLC
007654  006002                        ROR     R2
007656  022737  000004  001166        CMP     #4,RS04DT       ;RS03LA
007664  001003                        BNE     2$              ;NO
007666  004737  011756                JSR     PC,RRR2         ;YES
007672  000410                        BR      1$              ;CONTINUE
007674  000241                2$:     CLC
007676  006002                        ROR     R2
007700  004737  011756                JSR     PC,RRR2
007704  005737  001166                TST     RS04DT          ;RS04?
007710  001401                        BEQ     1$              ;NO
007712  006002                        ROR     R2              ;YES
007714  010237  001154        1$:     MOV     R2,BLOCK        ;BLOCK COUNT
007720  000207                        RTS     PC              ;EXIT

                                        ;ROUTINE TO SELECT DATA PATTERNS FOR TEST
                                        ;ENTER FROM JSR R5,PASEL
007722  012737  010356  000004  PASEL:  MOV     #MEM,@#4        ;SETUP TRAP
007730  012737  000340  000006        MOV     #340,@#6        ;VECTOR
007736  013700  001136                MOV     PATNO,R0        ;SET UP PATTERN NUMBER
007742  010003                        MOV     R0,R3           ;GET PATTERN #
007744  000241                        CLC                     ;MAKE IT =
007746  006003                        ROR     R3              ;TO PATTERN # IN LISTING
007750  010377  171054                MOV     R3,@DISPLAY     ;DISPLAY PATTERN #
007754  013737  001130  001222        MOV     WRDCT,WORK      ;SET UP WORK
007762  013701  001116                MOV     VADDR,R1        ;LOC. OF OUTBUFFER
007766  022700  000042        1$:     CMP     #42,R0          ;TEST FOR RANDOM DATA NUMBER
007772  001424                        BEQ     RANDOM          ;GO GENERATE RANDOM DATA
007774  032737  000004  001126        BIT     #BIT2,FLAG      ;MAX TST?
010002  001404                        BEQ     2$              ;NO
010004  016037  000254  017436        MOV     PAT0(0),OUTBUF  ;GET PATTERN
010012  000205                        RTS     R5
010014  016000  000254        2$:     MOV     PAT0(0),R0
010020  010021                FILDAT: MOV     R0,(1)+         ;FILL BUFFER
010022  005337  001222                DEC     WORK            ;DEC. WORK COUNT
010026  001374                        BNE     FILDAT          ;LOAD NEXT WORD
010030  012737  000006  000004  PASEX:  MOV     #6,@#4          ;RESTORE
010036  005037  000006                CLR     @#6             ;TRAP
010042  000205                        RTS     R5              ;BUFFER FULL
```

```
                                        ;RANDOM DATA GENERATOR SUBROUTINE
1892
1893
1894  010044  013737  010210  010214  RANDOM: MOV     LONUM,LOSAV
1895  010052  013737  010212  010216          MOV     HINUM,HISAV
1896  010060  013700  010210          RAND1:  MOV     LONUM,R0        ;SET UP R0 WITH 5 DIGITS LOW
1897  010064  013704  010212                  MOV     HINUM,R4        ;SET UP R1 WITH 5 DIGITS HIGH
1898  010070  012703  000007                  MOV     #7,R3           ;SET UP SHIFT COUNT
1899  010074  005002                          CLR     R2              ;CLEAR R2
1900  010076  006300                  SHIFT:  ASL     R0              ;SHIFT R0 LEFT AND
1901  010100  006104                          ROL     R4              ;ROTATE CARRY INTO LSB OF R1 INTO
1902  010102  006102                          ROL     R2              ;ROTATE CARRY OUT OF R1 INTO R2
1903  010104  005303                          DEC     R3              ;DECREMENT R3
1904  010106  001373                          BNE     SHIFT           ;CONTINUE SHIFT LOOP
1905  010110  063700  010210                  ADD     LONUM,R0        ;ADDN IN NUMBER TO MAKE * 123
1906  010114  005504                          ADC     R4              ;PROPOGATE CARRY
1907  010116  063704  010212                  ADD     HINUM,R4        ;ADDN IN NUMBER TO MAKE   123
1908  010122  005502                          ADC     R2              ;PROPOGATE CARRY
1909  010124  062700  001057                  ADD     #1057,R0        ;ADDN LOW CONSTANT
1910  010130  005504                          ADC     R4              ;PROPOGATE CARRIES
1911  010132  005502                          ADC     R2              ;PROPOGATE AGAIN
1912  010134  062704  047401                  ADD     #47401,R4       ;ADDN HIGH CONSTANT
1913  010140  005502                          ADC     R2              ;PROPOGATE CARRY
1914  010142  062702  000006                  ADD     #6,R2           ;ADDN HIGHEST CONSTANT
1915  010146  062700  000002                  ADD     #2,R0           ;REPRIME R0 WITH HIGH DIGIT
1916  010152  005504                          ADC     R4              ;PROPOGATE CARRY
1917  010154  010037  010210                  MOV     R0,LONUM        ;PUT R0 BACK IN LONUM
1918  010160  010021                          MOV     R0,(1)+         ;HOLD LONUM FOR PROGRAM
1919  010162  005337  001222                  DEC     WORK
1920  010166  001406                          BEQ     EXGEN
1921  010170  010437  010212                  MOV     R4,HINUM        ;PUT R1 BACK IN HINUM
1922  010174  010421                          MOV     R4,(1)+         ;HOLD HINUM FOR PROGRAM
1923  010176  005337  001222                  DEC     WORK
1924  010202  001326                          BNE     RAND1
1925  010204  000137  010070          EXGEN:  JMP     PASEX           ;RETURN TO PROGRAM
1926  010210  000000                  LONUM:  0
1927  010212  000000                  HINUM:  0
1928  010214  000000                  LOSAV:  0
1929  010216  000000                  HISAV:  0
1930
1931  010220  013737  001214  001144  RESTOR: MOV     SAVWC,SWRDCT    ;RESTORE ORIGINAL
1932  010226  013737  001144  001130          MOV     SWRDCT,WRDCT    ;WORD COUNT
1933  010234  013737  001220  001112          MOV     SAVWCB,WDCTB
1934  010242  000205                          RTS     R5
```

```
                                        ;RANDOM DATA GENERATOR SUBROUTINE
                                        ;WHEN SWITCH = 0 WE COME HERE
                                        
010244  013700  010352        RAND:   MOV     LONUM1,R0       ;SET UP R0 WITH 5 DIGITS LOW
010250  013704  010354                MOV     HINUM1,R4       ;SET UP R1 WITH 5 DIGITS HIGH
010254  012703  000007                MOV     #7,R3           ;SET UP SHIFT COUNT
010260  005002                        CLR     R2              ;CLEAR R2
010262  006300                SHIFT1: ASL     R0              ;SHIFT R0 LEFT AND
010264  006104                        ROL     R4              ;ROTATE CARRY INTO LSB OF R1 INTO
010266  006102                        ROL     R2              ;ROTATE CARRY OUT OF R1 INTO R2
010270  005303                        DEC     R3              ;DECREMENT R3
010272  001373                        BNE     SHIFT1          ;CONTINUE SHIFT LOOP
010274  063700  010352                ADD     LONUM1,R0       ;ADDN IN NUMBER TO MAKE * 129
010300  005504                        ADC     R4              ;PROPOGATE CARRY
010302  063704  010354                ADD     HINUM1,R4       ;ADDN IN NUMBER TO MAKE * 129
010306  005502                        ADC     R2              ;PROPOGATE CARRY
010310  062700  001057                ADD     #1057,R0        ;ADDN LOW CONSTANT
010314  005504                        ADC     R4              ;PROPOGATE CARRIES
010316  005502                        ADC     R2              ;PROPOGATE AGAIN
010320  062704  047401                ADD     #47401,R4       ;ADDN HIGH CONSTANT
010324  005502                        ADC     R2              ;PROPOGATE CARRY
010326  062702  000006                ADD     #6,R2           ;ADDN HIGHEST CONSTANT
010332  062700  000002                ADD     #2,R0           ;REPRIME R0 WITH HIGH DIGIT
010336  005504                        ADC     R4              ;PROPOGATE CARRY
010340  010037  010352                MOV     R0,LONUM1       ;PUT R0 BACK IN LONUM
010344  010437  010354                MOV     R4,HINUM1       ;PUT R1 BACK IN HINUM
010350  000205                EXGEN1: RTS     R5              ;RETURN TO PROGRAM
010352  000000                LONUM1: 0
010354  000000                HINUM1: 0
                                        
                                        ;TRAP OUT ROUTINE WHEN CREATING DATA BUFFER
                                        
010356                          MEM:
010356  104402  010362                TYPE    .+2             ;.ASCIZ <15><12>"NXM PORT "
010376  005737  001114                TST     A0B1            ;FIND WHAT DATA BUFFER
010402  001004                        BNE     3$              ;BRANCH IF B
010404  104402  010410                TYPE    .+2             ;.ASCIZ "A"
010412  000403                        BR      4$
010414                          3$:
010414  104402  010420                TYPE    .+2             ;.ASCIZ "B"
010422  012737  000006  000004  4$:   MOV     #6,@#4          ;RESTORE
010430  005037  000006                CLR     @#6             ;TRAP
010434  032777  100000  170364        BIT     #BIT15,@SWR     ;HALT?
010442  001401                        BEQ     2$              ;NO
010444  000000                        HALT
010446  000137  001234          2$:   JMP     @#BEGIN
```

```
                                        ;THIS ROUTINE COMPARES THE DATA READ AGAINST THE DATA EXPECTED.
                                        ;ALL ERRORS ARE REPORTED TO THE OPERATOR. IF BIT 4 OF THE SWITCH
                                        ;REGISTER IS SET, THIS ROUTINE WILL CONTINUE COMPARING AFTER AN ERROR HAS BEEN
                                        ;FOUND AND WILL REPORT UP TO 8 VERIFY ERRORS WITHIN THE SAME INPUT OPERATION.

010452  012737  177770  001146  COMPAR: MOV     #-10,ERCOUNT    ;ERROR RETRY COUNTER
010460  052737  000010  001122          BIS     #BIT3,FLAG2     ;DOING COMPARE
010466  013737  001130  001226          MOV     WRDCT,WORK2     ;GET THE WORD COUNT
010474  013737  001116  001150          MOV     VADDR,SAVE      ;SET UP OUTBUFFER POINTER
010502  005037  001174                  CLR     SWITCH          ;CLEAR RANDOM PATTERN FLAG
010506  013737  010214  010352          MOV     LOSAV,LONUM1    ;GET RANDOM BASE NOS.
010514  013737  010216  010354          MOV     HISAV,HINUM1
010522  005737  001136                  TST     PATNU           ;TEST FOR PATTERN 0
010526  001015                          BNE     1$              ;NO
010530  005037  001222                  CLR     WORK            ;CLEAR COUNTER
010534  062737  000001  001222  2$:     ADD     #1,WORK         ;INC COUNTER
010542  001003                          BNE     3$              ;INTERRUPT YET?
010544  104054                          HLT     !DA!WC!DS       ;TIMED OUT NOT INTERRUPT
010546  000137  001234                  JMP     @#BEGIN
010552  005737  001176          3$:     TST     INTFLG          ;TEST FOR INT
010556  001766                          BEQ     2$              ;WAIT FOR INT BEFORE COMPARING
010560  000426                          BR      CMPLP1          ;CONT
010562  022737  000042  001136  1$:     CMP     #42,PATNU       ;IS THIS RANDOM PATTERN?
010570  001022                          BNE     CMPLP1          ;BRANCH IF YES
010572  005737  001176          CMPLP:  TST     INTFLG          ;INTERRUPT YET?
010576  001775                          BEQ     CMPLP           ;NO WAIT
010600  005737  001174                  TST     SWITCH
010604  001007                          BNE     2$
010606  004537  010244                  JSR     R5,RAND
010612  013701  010352                  MOV     LONUM1,GOOD     ;GET EVEN RANDOM WORD
010616  010637  001174                  MOV     SP,SWITCH       ;SET RANDOM PATTERN FLAG
010622  000411                          BR      WRDCMP
010624  005037  001174          2$:     CLR     SWITCH
010630  013701  010354                  MOV     HINUM1,GOOD
010634  000404                          BR      WRDCMP
010636  013700  001136          CMPLP1: MOV     PATNU,R0
010642  016001  000254                  MOV     PATO(R0),GOOD
010646  160177  170276          WRDCMP: SUB     GOOD,@SAVE      ;COMPARE DATA
010652  001017                          BNE     WDERR           ;WORD IN ERROR
010654  005337  001226          WRDINC: DEC     WORK2           ;DECREMENT THE WORD COUNT
010660  001410                          BEQ     ADAM            ;EXIT ROUTINE IF ZERO
010662  062737  000002  001150          ADD     #2,SAVE         ;UPDATE PATTERN ADDRESS
010670  022737  000042  001136          CMP     #42,PATNU       ;IS THIS RANDOM PATTERN
010676  001735                          BEQ     CMPLP           ;BRANCH IF YES
010700  000762                          BR      WRDCMP          ;COMPARE NEXT WORD
010702  042737  000010  001122  ADAM:   BIC     #BIT3,FLAG2     ;DONE WITH COMPARE
010710  000205                          RTS     R5              ;EXIT THIS ROUTINE
```

```
2028  010712  005737  001176          WDERR:  TST     INTFLG          ;DID INTERRUPT OCCUR YET?
2029  010716  001753                          BEQ     WRDCMP          ;BRANCH IF NO
2030  010720  032777  000100  170100          BIT     #BIT6,@SWR      ;TRY ALL?
2031  010726  001006                          BNE     10$             ;YES
2032  010730  005737  001200                  TST     LOPCNT          ;FIRST READ ERROR?
2033  010734  001403                          BEQ     10$             ;YES
2034  010736  005777  170070                  TST     @RS0S1          ;ANY ERRORS?
2035  010742  100757                          BMI     ADAM            ;YES DO NOT COMPARE
2036  010744  060177  170200          10$:    ADD     GOOD,@SAVE
2037  010750  017700  170174                  MOV     @SAVE,BAD       ;GET GOOD DATA
2038  010754  104436                          LOGC                    ;LOG COMPARE ERROR
2039  010756  032777  001000  170042          BIT     #BIT9,@SWR      ;LOOP ON ERROR?
2040  010764  001401                          BEQ     11$             ;NO
2041  010766  005726                          TST     (6)+            ;YES UPDATE SP
2042  010770  004737  014224          11$:    JSR     PC,PRNT         ;TYPEOUT?
2043  010774  001007                          BNE     3$              ;NO
2044  010776  104402  011002                  TYPE    ,.+2            ;.ASCIZ <15><12>"CMP ERR"
2045  011014  104000                  3$:     HLT                     ;DATA COMPARE ERROR
2046  011016  004737  014224                  JSR     PC,PRNT         ;HAD TO DO IT THIS WAY SO
2047  011022  001022                          BNE     13$             ;PROGRAM COULD LOOP ON ERROR
2048  011024  104402  011030                  TYPE    ,.+2            ;.ASCIZ " ADDR="
2049  011040  005737  001212                  TST     MMAVA           ;IS MEM MGMT ON?
2050  011044  001406                          BEQ     12$             ;NO
2051  011046  013746  177776                  MOV     PS,-(6)         ;GET PS
2052  011052  013746  001150                  MOV     SAVE,-(6)       ;GET VIRTUAL ADDR
2053  011056  104412                          TYPEA                   ;CONVERT TO PHY AND TYPE
2054  011060  000403                          BR      13$             ;CONT
2055  011062  013746  001150          12$:    MOV     SAVE,-(6)       ;GET ADDR
2056  011066  104406                          TYPES                   ;TYPE IT
2057  011070  005037  001154          13$:    CLR     BLOCK           ;CLEAR THE BLOCK COUNTER
2058  011074  013702  001130                  MOV     WRDCT,R2        ;GET THE WORD COUNT
2059  011100  005202                          INC     R2              ;CORRECT FOR DA CALCULATIONS
2060  011102  163702  001226                  SUB     WORK2,R2        ;DETERMINE DISTANCE OF FAILURE INTO BUFFER
2061  011106  022737  000004  001166  2$:     CMP     #4,RS04DT       ;RS03LA?
2062  011114  001003                          BNE     14$             ;NO
2063  011116  162702  000040                  SUB     #40,R2
2064  011122  000410                          BR      9$              ;CONTINUE
2065  011124  005737  001166          14$:    TST     RS04DT          ;RS04?
2066  011130  001403                          BEQ     7$              ;NO
2067  011132  162702  000200                  SUB     #200,R2         ;RS03
2068  011136  000402                          BR      9$              ;CONT
2069  011140  162702  000100          7$:     SUB     #100,R2
2070  011144  100403                  9$:     BMI     8$
2071  011146  005237  001154                  INC     BLOCK           ;UPDATE BLOCK COUNT FOR EACH 400 WORDS
2072  011152  000755                          BR      2$
```

```
2073  011154  022737  000004  001166  8$:     CMP     #4,RS04DT       ;RS04LA
2074  011162  001003                          BNE     15$             ;NO
2075  011164  062702  000040                  ADD     #40,R2          ;RESTORE POSITIVE #
2076  011170  000410                          BR      6$              ;CONTINUE
2077  011172  005737  001166          15$:    TST     RS04DT          ;RS04?
2078  011176  001403                          BEQ     4$              ;NO
2079  011200  062702  000200                  ADD     #200,R2         ;RS04
2080  011204  000402                          BR      6$              ;CONT
2081  011206  062702  000100          4$:     ADD     #100,R2         ;RESTORE POSITIVE NUMBER
2082  011212  013737  001134  001224  6$:     MOV     DMA,WORK1       ;GET HEAD AND SECTOR ADDRESS
2083  011220  063737  001154  001224  5$:     ADD     BLOCK,WORK1
2084  011226  004737  014224                  JSR     PC,PRNT         ;TYPEOUT?
2085  011232  001014                          BNE     1$              ;NO
2086  011234  104402  011240                  TYPE    .+2             ;.ASCIZ " DA="
2087  011246  013746  001224                  MOV     WORK1,-(6)      ;PUT WORK1 ON STACK
2088  011252  104406                          TYPES                   ;TYPE STACK IN OCTAL - SL RESS
2089  011254  104402  011260                  TYPE    .+2             ;.ASCIZ <15><12>
2090  011264  032777  000020  167534  1$:     BIT     #BIT4,@SWR      ;RETRY?
2091  011272  001405                          BEQ     CLEAR           ;NO
2092  011274  005237  001146                  INC     ERCOUNT         ;UPDATE ERROR COUNTER
2093  011300  001402                          BEQ     CLEAR
2094  011302  000137  010654                  JMP     WRDINC
2095  011306  032737  000004  001126  CLEAR:  BIT     #BIT2,FLAG      ;XFER TEST?
2096  011314  001404                          BEQ     3$              ;NO
2097  011316  032737  010000  001126          BIT     #BIT12,FLAG     ;READ?
2098  011324  001412                          BEQ     2$              ;NO
2099  011326  013700  001116          3$:     MOV     VADDR,R0        ;GET STARTING ADD OF BUFFER
2100  011332  013701  001130                  MOV     WRDCT,R1        ;NOW
2101  011336  005020                  1$:     CLR     (R0)+           ;CLEAR BUFFER
2102  011340  005301                          DEC     R1              ;COUNT LOCATIONS
2103  011342  001375                          BNE     1$              ;WAIT TILL DONE
2104  011344  042737  000010  001122          BIC     #BIT3,FLAG2     ;DONE WITH COMPARE
2105  011352  000205                  2$:     RTS     R5              ;NOW GET OUT
2106
2107  011354  013737  001072  017436  APORT:  MOV     SAVAST,OUTBUF   ;SET STARTING ADDR FOR OUTBUF
2108  011362  013737  001072  001116          MOV     SAVAST,VADDR    ;SAVE OUTBUF ADDR
2109  011370  005737  001212                  TST     MMAVA           ;MEM MGMT?
2110  011374  001411                          BEQ     EXTT            ;NO
2111  011376  013702  001100                  MOV     SAVMGA,R2       ;SET UP MEM MGMT
2112  011402  004737  012040          MMSET:  JSR     PC,STMM2        ;SETUP MEM MGMT
2113  011406  010237  001116                  MOV     R2,VADDR
2114  011412  013737  001120  017436          MOV     PHADDR,OUTBUF
2115  011420  000207                  EXTT:   RTS     PC
2116
2117  011422  013737  001112  001130  BPORT:  MOV     WDCTB,WRDCT     ;GET WC FOR B PORT
2118  011430  013737  001076  017436          MOV     SAVCPU,OUTBUF
2119  011436  013737  001076  001116          MOV     SAVCPU,VADDR
2120  011444  005737  001212                  TST     MMAVA           ;MEM MGMT AVAILABLE?
2121  011450  001763                          BEQ     EXTT            ;NO
2122  011452  013702  001104                  MOV     SAVMGC,R2
2123  011456  000751                          BR      MMSET
```

```
2124                                    ;TYPE CAN NOT WRITE BLOCK
2125
2126  011460  004737  014224    WTNO:   JSR     PC,PRNT         ;TYPEOUT?
2127  011464  001001                    BNE     1$              ;NO
2128  011466  000000                    HALT                    ;HALT CANT WRITE BLOCK
2129  011470  005037  001200    1$:     CLR     LOPCNT          ;CLEAR ERR COUNTER
2130  011474  000207                    RTS     PC
2131
2132                                    ;ROUTINE TO SET UP STARTING ADDRESS FOR ALL PORTS
2133                                    ;AND TO CREATE WORD COUNT MAX= 20K
2134
2135  011476  013702  001070    EXTMEM: MOV     STAMEM,R2       ;GET BANK #
2136  011502  005702                    TST     R2              ;DID  HE TYPE 0?
2137  011504  001001                    BNE     3$              ;NO
2138  011506  005202                    INC     R2              ;YES MAKE 1
2139  011510  005737  001212    3$:     TST     MMAVA
2140  011514  001021                    BNE     1$              ;BRANCH IF MEM MGMT AVAILABLE
2141  011516  000241                    CLC
2142  011520  004737  011756          JSR     PC,RRR2
2143  011524  010237  001072          MOV     R2,SAVAST       ;SAVE A STARTIND ADDR
2144  011530  032737  000100  001126  BIT     #BIT6,FLAG      ;IS THERA A B PORT?
2145  011536  001430                    BEQ     2$              ;NO
2146  011540  013702  001074          MOV     STBCOM,R2       ;YES GET STARTING
2147  011544  000241                    CLC
2148  011546  004737  011756          JSR     PC,RRR2
2149  011552  010237  001076          MOV     R2,SAVCPU       ;SAVE IT
2150  011556  000420                    BR      2$              ;GET WC
2151  011560  000302          1$:     SWAB    R2
2152  011562  006002                    ROR     R2
2153  011564  010237  001100          MOV     R2,SAVMGA       ;SAVE ADDR FOR A PORT
2154  011570  032737  000100  001126  BIT     #BIT6,FLAG      ;IS THERE B PORT?
2155  011576  001410                    BEQ     2$              ;NO
2156  011600  013702  001074          MOV     STBCOM,R2
2157  011604  000302                    SWAB    R2
2158  011606  006002                    ROR     R2
2159  011610  010237  001102          MOV     R2,SAVMGB       ;SAVE B STARTING ADDR
2160  011614  010237  001104          MOV     R2,SAVMGC       ;SAVE CPU STARTING ADDR
2161  011620  013702  001106    2$:     MOV     SIZEAP,R2       ;GET 4K BLOCK COUNT
2162  011624  005202                    INC     R2
2163  011626  013703  001070          MOV     STAMEM, R3
2164  011632  160302                    SUB     R3,     R2
2165  011634  022702  000007          CMP     #7,R2   ;IS IT GREATER THEN 20K?
2166  011640  101411                    BLOS    4$              ;YES MAKE IT 20K
```

```
2167  011642  000241                    8$:   CLC
2168  011644  006002                          ROR   R2                ;NO CONVERT TO WC
2169  011646  004737  011756                  JSR   PC,RRR2
2170  011652  042702  000077                  BIC   #77,R2            ;CLEAR BLOCK COUNT
2171  011656  010237  001144                  MOV   R2,$WRDCT         ;SAVE -A- PORT WC
2172  011662  000403                          BR    5$                ;CONT
2173  011664  012737  060000  001144    4$:   MOV   #60000,$WRDCT     ;MAKE 20K
2174  011672  032737  000100  001126    5$:   BIT   #BIT6,FLAG        ;IS THERE B PORT
2175  011700  001425                          BEQ   7$                ;NO GET OUT
2176  011702  013702  001110                  MOV   SIZEBP,R2         ;GET B 4K COUNT
2177  011706  005202                          INC   R2
2178  011710  013703  001074                  MOV   STBCOM, R3
2179  011714  160302                          SUB   R3,     R2
2180  011716  022702  000007                  CMP   #7,R2   ;IS IT GREATER THEN 20K?
2181  011722  101411                          BLOS  6$                ;YES MAKE 20K
2182  011724  000241                    9$:   CLC
2183  011726  006002                          ROR   R2                ;NO CONVERT TO WC
2184  011730  004737  011756                  JSR   PC,RRR2
2185  011734  042702  000077                  BIC   #77,R2            ;CLEAR SECTOR COUNT
2186  011740  010237  001112                  MOV   R2,WDCTB          ;SAVE WC FOR -B- PORT
2187  011744  000403                          BR    7$                ;GET OUT
2188  011746  012737  060000  001112    6$:   MOV   #60000,WDCTB      ;MADE 20K WD
2189  011754  000207                    7$:   RTS   PC
2190
2191  011756  006002                    RRR2: ROR   R2
2192  011760  006002                          ROR   R2
2193  011762  006002                          ROR   R2
2194  011764  006002                          ROR   R2
2195  011766  000207                          RTS   PC
2196
2197  011770  032777  000200  167030    WATT: BIT   #BIT7,@SWR        ;WAIT IN BACKGROUND?
2198  011776  001003                          BNE   1$                ;NO
2199  012000  004737  012332                  JSR   PC,XWAIT          ;YES
2200  012004  000401                          BR    2$                ;CONT
2201  012006  000001                    1$:   WAIT
2202  012010  000207                    2$:   RTS   PC
2203
2204  012012  004737  014224          TYPREC: JSR   PC,PRNT          ;TYPEOUT
2205  012016  001007                          BNE   1$                ;NO
2206  012020  104402  000616                  TYPE  ,RECOV
2207  012024  013746  001200                  MOV   LOPCNT,-(6)       ;GET COUNT
2208  012030  104406                          TYPES                   ;TYPE IT
2209  012032  104402  000636                  TYPE  ,CRLFLF
2210  012036  000207                    1$:   RTS   PC
```

```
2211
2212  012040  005737  001212          STMM2:  TST     MMAVA           ;MEM MGMT?
2213  012044  001002                          BNE     3$              ;YES
2214  012046  000137  012324                  JMP     MDON            ;GET OUT
2215  012052  005037  172340          3$:     CLR     @#KIPAR0
2216  012056  010237  001150                  MOV     R2,SAVE         ;SAVE R2
2217  012062  010237  172342                  MOV     R2,@#KIPAR1
2218  012066  006302                          ASL     R2              ;CALCULATE PHYSICAL ADDR
2219  012070  006302                          ASL     R2
2220  012072  006302                          ASL     R2
2221  012074  006302                          ASL     R2
2222  012076  006302                          ASL     R2              ;THIS BIT IS A17
2223  012100  042737  000040  001122        BIC     #BIT5,FLAG2     ;CLEAR A17?
2224  012106  103003                          BCC     1$              ;SET A17
2225  012110  052737  000040  001122        BIS     #BIT5,FLAG2     ;SET BIT 5 FOR A17
2226  012116  042737  000020  001122  1$:     BIC     #BIT4,FLAG2     ;CLEAR A16 FLAG
2227  012124  006302                          ASL     R2              ;GET A16 BIT
2228  012126  103003                          BCC     2$              ;CLEAR A16
2229  012130  052737  000020  001122        BIS     #BIT4,FLAG2     ;SET FLAG FOR A16
2230  012136  010237  001120          2$:     MOV     R2,PHADDR       ;GET PHYSICAL ADDR
2231  012142  013702  001150                  MOV     SAVE,R2         ;SET UP MEM MGMT
2232  012146  062702  000200                  ADD     #200,R2
2233  012152  010237  172344                  MOV     R2,@#KIPAR2
2234  012156  062702  000200                  ADD     #200,R2
2235  012162  010237  172346                  MOV     R2,@#KIPAR3
2236  012166  062702  000200                  ADD     #200,R2
2237  012172  010237  172350                  MOV     R2,@#KIPAR4
2238  012176  062702  000200                  ADD     #200,R2
2239  012202  010237  172352                  MOV     R2,@#KIPAR5
2240  012206  062702  000200                  ADD     #200,R2
2241  012212  010237  172354                  MOV     R2,@#KIPAR6
2242  012216  012737  077406  172300          MOV     #200*256.-400+UP+RW,@#KIPDR0    ;SET KIPDR0=RW UP 200 BLOCKS
2243  012224  012737  077406  172302          MOV     #200*256.-400+UP+RW,@#KIPDR1    ;SET KIPDR1=RW UP 200 BLOCKS
2244  012232  012737  077406  172304          MOV     #200*256.-400+UP+RW,@#KIPDR2    ;SET KIPDR2=RW UP 200 BLOCKS
2245  012240  012737  077406  172306          MOV     #200*256.-400+UP+RW,@#KIPDR3    ;SET KIPDR3=RW UP 200 BLOCKS
2246  012246  012737  077406  172310          MOV     #200*256.-400+UP+RW,@#KIPDR4    ;SET KIPDR4=RW UP 200 BLOCKS
2247  012254  012737  077406  172312          MOV     #200*256.-400+UP+RW,@#KIPDR5    ;SET KIPDR5=RW UP 200 BLOCKS
2248  012262  012737  077406  172314          MOV     #200*256.-400+UP+RW,@#KIPDR6    ;SET KIPDR6=RW UP 200 BLOCKS
2249  012270  012737  077406  172316          MOV     #200*256.-400+UP+RW,@#KIPDR7    ;SET KIPDR7=RW UP 200 BLOCKS
2250  012276  012737  007600  172356          MOV     #7600,@#KIPAR7
2251  012304  012702  020000                  MOV     #20000,R2
2252  012310  012737  012326  000250          MOV     #MMABTO,@#MMVEC
2253  012316  012737  000001  177572          MOV     #1,@#SR0        ;TURN ON MEM MGMT
2254  012324  000207                  MDON:   RTS     PC
2255
2256                                  ;MEMORY MANAGEMENT ABORT ROUTINE FOR WRITE UP
2257  012326  000000                  MMABTO: HALT                    ;MEMORY MANAGEMENT TRAP
2258  012330  000002                          RTI                     ;CAUSED THE ABORT
```

```
                                        ;BACKGROUND TEST FOR INTERRUPTS
                                        
012332  052737  002000  001122  \WAIT:  BIS     #BIT10,FLAG2    ;WAITING IN BACKGROUND TEST
012340  012737  070000  012446          MOV     #70000,NPRCNT   ;SETUP TIMEOUT COUNTER
012346  012701  012451                  MOV     #NPR1+1,R1      ;SETUP WAIT LOOP
012352  112711  000200                  MOVB    #200,(R1)
012356                          2$:
012356  105421                          NEGB    (R1)+
012360  105441                          NEGB    -(R1)
012362  105421                          NEGB    (R1)+
012364  105441                          NEGB    -(R1)
012366  105421                          NEGB    (R1)+
012370  105441                          NEGB    -(R1)
012372  105421                          NEGB    (R1)+
012374  105441                          NEGB    -(R1)
012376  105421                          NEGB    (R1)+
012400  105441                          NEGB    -(R1)
012402  105421                          NEGB    (R1)+
012404  105441                          NEGB    -(R1)
012406  105421                          NEGB    (R1)+
012410  105441                          NEGB    -(R1)
012412  105421                          NEGB    (R1)+
012414  105441                          NEGB    -(R1)
012416  102401                          BVS     1$
012420  000000                          HALT                    ;ARITHMETIC OPERATION FAILED RUN DIAG
012422  005337  012446          1$:     DEC     NPRCNT
012426  001353                          BNE     2$
012430  104054                          HLT     !DA!WC!DS       ;TIMED OUT NO INTERRUPT
012432  000137  001234                  JMP     @#BEGIN
012436  042737  002000  001122  NPRRET: BIC     #BIT10,FLAG2    ;CLEAR BKGROUND FLG
012444  000207                          RTS     PC
012446  000000                  NPRCNT: 0
012450  000000                  NPR1:   0
                                        ;CLEAR ERROR TABLE
                                        
012452  012704  000020          ERRCL:  MOV     #20,R4          ;CLEAR
012456  012703  017072                  MOV     #ERTAB,R3       ;ERROR
012462  005023                  1$:     CLR     (R3)+           ;TABLE
012464  005304                          DEC     R4              ;DONE YET?
012466  001375                          BNE     1$              ;NO
012470  005037  001004                  CLR     PCNT            ;CLEAR
012474  005037  001006                  CLR     PCNT+2          ;PASS COUNT
012500  005037  001124                  CLR     DROP            ;CLEAR ALL DROPPED DRIVES
012504  000205                          RTS     R5              ;RETURN
```

```
                                        ;RH11 POWER FAIL TEST #1
                                        ;THE STARTING ADDRESS FOR THE WRITE POWER FAIL TEST IS 244. THE PROGRAM
                                        ;WRITE THE COMPLETE DISK WITH A 125252 PATTERN. THE PROGRAM WILL THEN
                                        ;TELL OPERATOR TO POWER DOWN. UNTIL THE POWER FAIL, THE PROGRAM WILL
                                        ;CONTINUE WRITING THE SAME PATTERN ON THE DISK.
                                        ;WHEN POWER FAIL OCCURS THE TRANSFER IS ABORTED
                                        ;AND THE PROGRAM HALTS. THE OPERATOR SHOULD
                                        ;NOW TURN POWER BACK ON. THE PROGRAM RESTARTS AND CHECKS FOR WRITE ERRORS.
                                        ;ONLY ONE ERROR IS ACCEPTABLE. THAT ERROR MAY BE AN OPI (BIT13 RSER OR A DCK
                                        ; BIT 15 RSER). IF THESE ARE THE ONLY ERRORS THAT OCCUR, THE DRIVE IS OK.
                                        ;IF NO ERRORS OCCUR, THE PROGRAM WILL TYPE OUT "OK".
                                        ;THE PROGRAM WILL THEN TELL YOU WHEN TO POWER DOWN AGAIN

                                        ;***ONLY ONE ERROR IS CONSIDERED ACCEPTABLE***
                                        ;NOTE: ALL DRIVES ON THE SYSTEM SHOULD BE POWERED OFF EXCEPT
                                        ;THE DRIVE UNDER TEST. ******************

012506  012706  000500          PFT1:   MOV     #500,SP         ;SET UP STACK
012512  104402  000510                  TYPE    ,LOADSW
012516  104420                          RDOCT
012520  012637  001160                  MOV     (SP)+,UNNUM
012524  004737  007000          2$:     JSR     PC,VECTRR       ;SETUP INT VECTOR
012530  004737  006544                  JSR     PC,FNDTYP       ;TST FOR RS03 OR 04
012534  104426                  PFWATT: CLRDV                   ;CLEAR ALL REG
012536  004737  013312                  JSR     PC,POWFAL       ;WRITE 125252 ON DISK
012542  005037  001134          PFWAT:  CLR     DMA
012546  012737  012770  000024        MOV     #DOWN,24        ;SET UP POWER FAIL VEC.
012554  012737  000340  000026        MOV     #340,26
012562  012737  000161  001172  MYBYWR. MOV     #161,CMD        ;WRITE WITH I/E
012570  104416                          DKCMD                   ;DO IT
012572  004737  011770                  JSR     PC,WATT         ;WAIT FOR INTERRUPT
012576  032737  001000  001126  3$:     BIT     #BIT9,FLAG      ;ANY ERRORS?
012604  001406                          BEQ     1$              ;NO
012606  104006                          HLT     !DA!DB
012610  012777  177777  166232        MOV     #-1,@RSAS       ;CLEAR ALL
012616  005077  166224                CLR     @RSER           ;ERRORS
012622  004737  007054          1$:     JSR     PC,DISBUF       ;SET UP NEW DISK BUFFER
012626  000755                          BR      MYBYWR
012630  000744                          BR      PFWAT
```

```
2342  012632  012737  012640  001152  UPCHK:  MOV     #1$,HRDER       ;RETURN HERE IF WRONG DRIVE INTERRUPTS
2343  012640  005037  001134          1$:     CLR     DMA
2344  012644  104426                          CLRDV                   ;INIT DRIVE
2345  012646  013737  001066  177776  CHKDAT: MOV     PRIORITY,PS
2346  012654  012737  000151  001172          MOV     #151,CMD        ;WRITECHECK WITH I E
2347  012662  104416                          DKCMD                   ;DO IT
2348  012664  013737  001066  177776          MOV     PRIORITY,PS
2349  012672  004737  011770                  JSR     PC,WAITT        ;WAIT FOR INTERPUPT
2350  012676  032737  001000  001126  3$:     BIT     #BIT9,FLAG      ;ANY ERRORS?
2351  012704  001411                          BEQ     1$              ;NO
2352  012706  104006                          HLT     !DB!DA
2353  012710  052737  100000  001122          BIS     #BIT15,FLAG2    ;SET ERROR FLAG
2354  012716  005077  166124                  CLR     @RSER           ;CLEAR ALL
2355  012722  012777  177777  166120          MOV     #-1,@RSAS       ;ERRORS
2356  012730  004737  007054          1$:     JSR     PC,DISBUF       ;SET UP NEW DISK BUFFER
2357  012734  000744                          BR      CHKDAT
2358  012736  005737  001122                  TST     FLAG2           ;ANY ERRORS?
2359  012742  100405                          BMI     2$              ;YES
2360  012744  104402  012750                  TYPE    ,.+2            ;.ASCIZ <15><12>"OK"
2361  012756  042737  100000  001122  2$:     BIC     #BIT15,FLAG2    ;CLEAR ERROR FLAG
2362  012764  000137  012534                  JMP     PFWAIT          ;GO WAIT FOR ANOTHER
2363                                                                  ;POWER FAIL
2364
2365                                  ;POWER DOWN ROUTINE - ABORT DISK AND HALT
2366
2367  012770  012737  013000  000024  DOWN:   MOV     #UPP,24         ;SET POWER FAIL VECTOR
2368  012776  000000                          HALT
2369
2370  013000  012737  012770  000024  UPP:    MOV     #DOWN,24
2371  013006  012706  000500                  MOV     #500,SP
2372  013012  013777  001160  166014          MOV     UNNUM,@RSCS2    ;GET UNIT #
2373  013020  032777  000200  166016  1$:     BIT     #BIT7,@RSDS     ;WAIT FOR DRIVE READY
2374  013026  001774                          BEQ     1$
2375  013030  000137  012632                  JMP     UPCHK           ;GO CHECK DISK
```

```
                                        ;POWER FAIL TEST #2
                                        ;THIS TEST WILL TEST THE SAME DRIVE THAT WAS TESTED IN THE 1ST POWER FAIL TEST
                                        ;THE PROGRAM WILL WRITE THE COMPLETE DISK WITH A 125252 PATTERN AND WILL
                                        ;THEN TELL THE OPERATOR TO POWER DOWN THE PROCESSOR.
                                        ;THE PROGRAM WILL THEN WRITE CHECK THE DISK WHILE WAITING FOR A POWER FAIL.
                                        ;WHEN THE POWER FAIL OCCURS, THE WRITE CHECKING IS ABORTED AND
                                        ;THE PROCESSOR WILL HALT.
                                        ;THE OPERATOR SHOULD THEN TURN POWER BACK ON, THE PROGRAM WILL
                                        ;START WRITE CHECKING THE DISK AGAIN
                                        ;***NO ERRORS SHOULD OCCUR.***
                                        ;THE PROGRAM WILL TYPE OUT "OK" IF NO ERRORS OCCUR.
                                        ;THE PROGRAM WILL THEN TELL YOU TO POWER DOWN.
                                        ;DO NOT POWER OFF THE PROCESSOR AGAIN UNTIL THE PROGRAM TELLS YOU SO.
                                        ;NOTE: ALL DRIVES ON THE SYSTEM SHOULD BE POWERED OFF
                                        ;EXCEPT THE ONE UNDER TEST ********************

013034  012706  000500          PFT2:   MOV     #500,SP         ;SET UP STACK
013040  042737  001000  001122          BIC     #BIT9,FLAG2     ;CLEAR POWER FAIL
013046  012737  013070  001152          MOV     #PWRF1,HRDER    ;RETURN HERE IF WRONG DRIVE INT.
013054  104426                          CLRDV                   ;INIT DRIVE
013056  004737  007000                  JSR     PC,VECTRR       ;SETUP INT VECTOR
013062  004737  013312          PWRFL2: JSR     PC,POWFAL       ;WRITE 125252 ON DISK
013066  000401                          BR      PWRFL           ;WRITE CHECK
013070  104426                  PWRF1:  CLRDV                   ;INIT DRIVE
013072  005037  001134          PWRFL:  CLR     DMA
013076  012737  013242  000024          MOV     #PWRDN,24       ;SET UP POWER FAIL VEC.
013104  012737  000340  000026          MOV     #340,26
013112  013737  001066  177776  CHKDSK: MOV     PRIORITY,PS     ;ENABLE I/E
013120  012737  000151  001172          MOV     #151,CMD        ;WRITE CHECK WITH I/E
013126  104416                          DKCMD                   ;DO IT
013130  004737  011770                  JSR     PC,WATT         ;WAIT FOR INTERRUPT
013134  032737  001000  001126  3$:     BIT     #BIT9,FLAG      ;ANY ERRORS?
013142  001411                          BEQ     1$              ;NO
013144  104002                          HLT     !DB             ;YES
013146  052737  100000  001122          BIS     #BIT15,FLAG2    ;SET ERROR FLAG
013154  005077  165666                  CLR     @RSER           ;CLEAR ALL
013160  012777  177777  165662          MOV     #-1,@RSAS       ;ERRORS
013166  004737  007054          1$:     JSR     PC,DISBUF       ;CHECK NEXT BUFFER
013172  000747                          BR      CHKDSK
013174  032737  001000  001122          BIT     #BIT9,FLAG2     ;DID POWER FAIL?
013202  001733                          BEQ     PWRFL           ;NO
013204  005737  001122                  TST     FLAG2           ;ANY ERRORS?
013210  100405                          BMI     2$              ;YES
013212  104402  013216                  TYPE    ,.+2            ;.ASCIZ <15><12>"OK"
013224  042737  100000  001122  2$:     BIC     #BIT15,FLAG2    ;CLEAR ERRORS
013232  042737  001000  001122          BIC     #BIT9,FLAG2     ;CLEAR POWER FAIL FLAG
013240  000710                  4$:     BR      PWRFL2
```

```
                                        ;ROUTINE TO ABORT DISK DURING POWER FAIL
                                        
013242  012737  013252  000024  PWRDN:  MOV     #PWRUP,24       ;SET UP RESTART
013250  000000                  HALT
                                        
013252  012737  013242  000024  PWRUP:  MOV     #PWRDN,24       ;RESET POWER FAIL VECTOR
013260  012706  000500          MOV     #500,SP
013264  013777  001160  165542          MOV     UNNUM,@RSCS2    ;GET UNIT #
013272  052737  001000  001122          BIS     #BIT9,FLAG2     ;SET POWER FAIL BIT
013300  032777  000200  165536  1$:     BIT     #BIT7,@RSDS     ;WAITING FOR
013306  001774                  BEQ     1$              ;DRIVE READY
013310  000667                  BR      PWRF1           ;GO CHECK DISK
                                        
                                        
                                        ;ROUTINE TO WRITE THE COMPLETE DISK
                                        ;WITH 125252 PATTERN
                                        ;WRITE CHECK AND REPORT ERRORS IF THEY  OCCUR
                                        ;REPORT "OK" AT COMPLETION
                                        
013312  012737  000020  001136  POWFAL: MOV     #20,PATNU       ;SET UP PATTERN
013320  042737  000004  001126          BIC     #BIT2,FLAG      ;CLEAR XFER MODE FLAG
013326  052737  010000  001122          BIS     #BIT12,FLAG2
013334  005037  001134          CLR     DMA
013340  012737  020000  017436          MOV     #20000,OUTBUF   ;GET STARTING ADDR FOR BUF
013346  012737  020000  001116          MOV     #20000,VADDR
013354  012737  010000  001144          MOV     #10000,SWRDCT   ;SETUP WORD COUNT
013362  013737  001144  001130          MOV     SWRDCT,WRDCT
013370  005037  001114          CLR     ADB1            ;A PORT ONLY
013374  013737  017436  001140          MOV     OUTBUF,BUF      ;SET UP CURRENT ADDRESS
013402  004537  007722          JSR     R5,PASEL        ;GENERATE DATA BUFFER
013406  012737  000161  001172  WRDNW:  MOV     #161,CMD        ;WRITE WITH I/E
013414  104416                  DKCMD                   ;DO IT
013416  004737  011770          JSR     PC,WATT         ;WAIT FOR INTERRUPT
013422  012737  000151  001172  2$:     MOV     #151,CMD        ;WRITECHECK I/E
013430  104416                  DKCMD                   ;DO IT
013432  004737  011770          JSR     PC,WATT         ;WAIT FOR INTERRUPT
013436  032737  001000  001126  4$:     BIT     #BIT9,FLAG      ;ANY ERRORS?
013444  001402                  BEQ     1$              ;NO
013446  104006                  HLT     !DB!DA          ;YES
013450  000000                  HALT                    ;CAN NOT WRITE WITHOUT ERROR
013452  004737  007054          1$:     JSR     PC,DISBUF       ;SET UP NEW DISK BUFFER
013456  000753                  BR      WRDNW           ;WRITE NEW BUFFER
013460  104402  000721          TYPE    ,PDOWN
013464  000207                  RTS     PC
```

```
 2467  013466  032777  000010  165332  OUT:    BIT     #BIT3,@SWR      ;TYPEOUT ERROR COUNT?
 2468  013474  001526                          BEQ     1$              ;NO
 2469  013476  005004                          CLR     R4              ;CLEAR UNIT #
 2470  013500  005003                          CLR     R3
 2471  013502  053737  001124  001162          BIS     DROP,UNITSV     ;RESTORE ALL DRIVES
 2472  013510  013737  001162  001222          MOV     UNITSV,WORK     ;GET UNITS ON SYSTEM
 2473  013516  012705  000401                  MOV     #40,R5          ;SETUP TEST FOR UNITS
 2474  013522  030537  001222          4$:     BIT     R5,WORK         ;IS THIS UNIT ON SYS
 2475  013526  001006                          BNE     2$              ;YES
 2476  013530  005204                  5$:     INC     R4              ;INC UNIT #
 2477  013532  010403                          MOV     R4,R3           ;SAVE UNIT #
 2478  013534  000241                          CLC
 2479  013536  006105                          ROL     R5              ;GET NEXT DRIVE
 2480  013540  103501                          BCS     3$              ;DONE
 2481  013542  000767                          BR      4$              ;FIND NEXT DRIVE
 2482  013544  104402  000510          2$:     TYPE    ,LOADSW
 2483  013550  010446                          MOV     R4,-(6)         ;PUT R4 ON STACK
 2484  013552  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
 2485  013554  004737  014032                  JSR     PC,GETERR       ;GET ERROR COUNT
 2486  013560  010304                          MOV     R3,R4           ;RESTORE UNIT #
 2487  013562  104402  013566                  TYPE    ,.+2            ;.ASCIZ <15><12>
 2488  013572  104402  000564                  TYPE    ,WRTERR
 2489  013576  104402  013602                  TYPE    ,.+2            ;.ASCIZ "S "
 2490  013606  013746  001202                  MOV     WRITER,-(6)     ;PUT WRITER ON STACK
 2491  013612  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
 2492  013614  104402  013620                  TYPE    ,.+2            ;.ASCIZ <15><12>
 2493  013624  104402  000607                  TYPE    ,RDERR
 2494  013630  104402  013634                  TYPE    ,.+2            ;.ASCIZ "S "
 2495  013640  013746  001206                  MOV     READER,-(6)     ;PUT READER ON STACK
 2496  013644  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
 2497  013646  104402  013652                  TYPE    ,.+2            ;.ASCIZ <15><12>
 2498  013656  104402  000574                  TYPE    ,WCKERR
 2499  013662  104402  013666                  TYPE    ,.+2            ;.ASCIZ "S "
 2500  013672  013746  001204                  MOV     WCERR,-(6)      ;PUT WCERR ON STACK
 2501  013676  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
 2502  013700  104402  013704                  TYPE    ,.+2            ;.ASCIZ <15><12>"COMPARE ERRS "
 2503  013724  013746  001210                  MOV     COMERR,-(6      ;PUT COMERR ON STACK
 2504  013730  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
 2505  013732  104402  013736                  TYPE    ,.+2            ;.ASCIZ <15><12>
 2506  013742  000672                          BR      5$              ;GET NEXT DRIVE
 2507  013744  043737  001124  001162  3$:     BIC     DROP,UNITSV     ;REDROP DRIVES
 2508  013752  062706  000002          1$:     ADD     #2,SP           ;RESTORE SP DUE TO JMP EXIT FROM JSR ROUTINE
 2509  013756  005137  001114                  COM     AOB1            ;SET A OR B PORT FLAG
 2510  013762  032777  000040  165036          BIT     #BIT5,@SWR      ;TYPEOUT PASS COUNT?
 2511  013770  001035                          BNE     DONE            ;NO
 2512  013772  104402  013776                  TYPE    ,.+2            ;.ASCIZ <15><12>"END PASS "
 2513  014012  013746  001006                  MOV     PCNT+2,-(6)     ;PUT PCNT+2 ON STACK
 2514  014016  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
 2515  014020  104402  014024                  TYPE    ,.+2            ;.ASCIZ <15><12>
 2516  014030  000415                          BR      DONE
```

```
2517  014032  006304                    GETERR: ASL     R4              ;GET LOC IN
2518  014034  006304                            ASL     R4              ;ERR TABLE
2519  014036  062704  017072                    ADD     #ERTAB,R4
2520  014042  112437  001202                    MOVB    (R4)+,WRITER    ;GET WRITE ERRS
2521  014046  112437  001206                    MOVB    (R4)+,READER    ;GET READ ERRS
2522  014052  112437  001204                    MOVB    (R4)+,WCERR     ;GET WRITE CK ERRS
2523  014056  112437  001210                    MOVB    (R4)+,COMERR    ;GET COMPARE ERRS
2524  014062  000207                            RTS     PC
2525
2526                                    .SBTTL          $DONE - BELL AND SCOPE ROUTINE
2527
2528  014064  104400                    DONE:   SCOPE                   ;TERMINATIONG SCOPE FOR LOOPING
2529  014066  062737  000001  001006            ADD     #1,PCNT+2       ;ADD 1 TO THE PASS COUNT
2530  014074  005537  001004                    ADC     PCNT            ;MAKE IT DOUBLE PREC.
2531  014100  013700  000042            4$:     MOV     @#42,R0         ;GET MONITOR ADDRESS
2532  014104  001405                            BEQ     $END1           ;IF NONE
2533  014106  000005                            RESET
2534  014110  004710                    $ENDAD: JSR     7,(0)           ;GO TO MONITOR
2535  014112  000240  000240  000240            240,240,240             ;SAVE ROOM FOR ACT11
2536  014120  000137  003232            $END1:  JMP     ADTST           ;RETURN
2537
2538  014124  000000                    .TBIT:  0                       ;T BIT FLAG
2539
2540  014126  012702  000001            .LOGW:  MOV     #1,R2           ;LOG WRITE ERR
2541  014132  005003                    CLIND:  CLR     R3              ;CLEAR INDEX FOR TABLE
2542  014134  000413                            BR      ADDRR
2543
2544  014136  012702  000400            .LOGR:  MOV     #400,R2         ;LOG WRITE ERR
2545  014142  000773                            BR      CLIND
2546
2547  014144  012702  000001            .LOGWC: MOV     #1,R2           ;LOG WRITE CK ERR
2548  014150  012703  000002            SETIND: MOV     #2,R3           ;SET INDEX FOR NEXT WD
2549  014154  000403                            BR      ADDRR
2550
2551  014156  012702  000400            .LOGC:  MOV     #400,R2         ;LOG COMPARE ERR
2552  014162  000772                            BR      SETIND
2553
2554  014164  005737  001200            ADDRR:  TST     LOPCNT          ;1ST TIME ERROR?
2555  014170  001014                            BNE     1$              ;NO DO NOT COUNT IT
2556  014172  013704  001160                    MOV     UNNUM,R4        ;GET UNIT #
2557  014176  006304                            ASL     R4              ;GET
2558  014200  006304                            ASL     R4              ;POSITION IN
2559  014202  060304                            ADD     R3,R4           ;ERR TABLE
2560  014204  060264  017072                    ADD     R2,ERTAB(R4)    ;TO ADD ERROR
2561  014210  004737  014224                    JSR     PC,PRNT         ;TYPEOUT?
2562  014214  001402                            BEQ     1$              ;YES
2563  014216  004737  015012                    JSR     PC,DRP          ;SHOULD I DROP DRIVE?
2564  014222  000002                    1$:     RTI
2565
2566  014224  032777  020000  164574    PRNT:   BIT     #BIT13,@SWR     ;INHIBIT TYPEOUT?
2567  014232  000207                            RTS     PC
```

```
2568  014234  052737  000004  001122  RSREG:  BIS     #BIT2,FLAG2     ;SET ERROR FLAG
2569  014242  005737  016060                  TST     .HLTCT          ;SHOULD WE TYPE GOOD AND BAD
2570  014246  001017                          BNE     8$              ;NO
2571  014250  104402  014254                  TYPE    .+2             ;.ASCIZ "BAD="
2572  014262  010046                          MOV     BAD,-(6)        ;PUT BAD ON STACK
2573  014264  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2574  014266  104402  014272                  TYPE    .+6             ;.ASCIZ " GOOD="
2575  014302  010146                          MOV     GOOD,-(6)       ;PUT GOOD ON STACK
2576  014304  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2577  014306                          8$:
2578  014306  104402  014312                  TYPE    .+2             ;.ASCIZ " CS1="
2579  014320  017746  164506                  MOV     @RSCS1,-(6)     ;PUT @RSCS1 ON STACK
2580  014324  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2581  014326                          1$:
2582  014326  104402  014332                  TYPE    .+2             ;.ASCIZ " ER="
2583  014340  017746  164502                  MOV     @RSER,-(6)      ;PUT @RSER ON STACK
2584  014344  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2585  014346                          2$:
2586  014346  104402  014352                  TYPE    .+2             ;.ASCIZ " CS2="
2587  014360  017746  164450                  MOV     @RSCS2,-(6)     ;PUT @RSCS2 ON STACK
2588  014364  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2589  014366  032737  000200  016060          BIT     #200,.HLTCT     ;PRINT SECOND SET ?
2590  014374  001112                          BNE     SEEC            ;YES
2591  014376  032737  000100  016060          BIT     #AS,.HLTCT      ;PRINT ER ?
2592  014404  001410                          BEQ     3$              ;NO
2593  014406  104402  014412                  TYPE    .+2             ;.ASCIZ " AS="
2594  014420  017746  164424                  MOV     @RSAS,-(6)      ;PUT @RSAS ON STACK
2595  014424  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2596  014426  032737  000020  016060  3$:     BIT     #BA,.HLTCT      ;PRINT BUS ASSRESS
2597  014434  001410                          BEQ     4$              ;NO
2598  014436  104402  014442                  TYPE    .+2             ;.ASCIZ " BA="
2599  014450  017746  164364                  MOV     @RSBA,-(6)      ;PUT @RSBA ON STACK
2600  014454  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2601  014456  032737  000004  016060  4$:     BIT     #DA,.HLTCT      ;PRINT DA ?
2602  014464  001410                          BEQ     5$              ;NO
2603  014466  104402  014472                  TYPE    .+2             ;.ASCIZ " DA="
2604  014500  017746  164336                  MOV     @RSDA,-(6)      ;PUT @RSDA ON STACK
2605  014504  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2606  014506  032737  000010  016060  5$:     BIT     #WC,.HLTCT      ;PRINT WC?
2607  014514  001410                          BEQ     6$              ;NO
2608  014516  104402  014522                  TYPE    .+2             ;.ASCIZ " WC="
2609  014530  017746  164302                  MOV     @RSWC,-(6)      ;PUT @RSWC ON STACK
2610  014534  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2611  014536  032737  000040  016060  6$:     BIT     #DS,.HLTCT      ;DRIVE STATUS
2612  014544  001410                          BEQ     9$              ;NO
2613  014546  104402  014552                  TYPE    .+2             ;.ASCIZ " DS="
2614  014560  017746  164260                  MOV     @RSDS,-(6)      ;PUT @RSDS ON STACK
2615  014564  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2616  014566  032737  000002  016060  9$:     BIT     #DB,.HLTCT      ;PRINT DATA BUFFER
```

```
2617  014574  001461                          BEQ     PTDONE          ;NO
2618  014576  104402  014602                  TYPE    .+2             ;.ASCIZ " DB="
2619  014610  017746  164240                  MOV     @RSDB,-(6)      ;PUT @RSDB ON STACK
2620  014614  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2621  014616  000137  014740                  JMP     PTDONE          ;GET OUT
2622  014622  042737  000200  016060  SEEC:   BIC     #200,.HLTCT     ;CLEAR COMMON BIT
2623  014630  032737  000240  016060          BIT     #DT,.HLTCT      ;PRINT DRIVE TYPE?
2624  014636  001410                          BEQ     10$             ;NO
2625  014640  104402  014644                  TYPE    .+2             ;.ASCIZ " DT="
2626  014652  017746  164202                  MOV     @RSDT,-(6)      ;PUT @RSDT ON STACK
2627  014656  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2628  014660  032737  000220  016060  10$:    BIT     #MR,.HLTCT      ;PRINT MN?
2629  014666  001410                          BEQ     11$             ;NO
2630  014670  104402  014674                  TYPE    .+2             ;.ASCIZ " MR="
2631  014702  017746  164150                  MOV     @RSMR,-(6)      ;PUT @RSMR ON STACK
2632  014706  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2633  014710  032737  000204  016060  11$:    BIT     #LA,.HLTCT      ;PRINT LA?
2634  014716  001410                          BEQ     PTDONE          ;NO
2635  014720  104402  014724                  TYPE    .+2             ;.ASCIZ " LA="
2636  014732  017746  164114                  MOV     @RSLA,-(6)      ;PUT @RSLA ON STACK
2637  014736  104404                          TYPEO                   ;TYPE STACK IN OCTAL
2638  014740  032737  010000  001122  PTDONE: BIT     #BIT12,FLAG2    ;POWER FAIL TEST?
2639  014746  001111                          BNE     RETT            ;YES
2640  014750  104402  014754                  TYPE    .+2             ;.ASCIZ <15><12>"PASS "
2641  014764  013746  001006                  MOV     PCNT+2,-(6)     ;PUT PCNT+2 ON STACK
2642  014770  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
2643  014772  032777  001000  164026          BIT     #BIT9,@SWR      ;LOOPING ON ERROR?
2644  015000  001404                          BEQ     DRP             ;NO
2645  015002  104402  015006                  TYPE    .+2             ;.ASCIZ <15><12>
2646  015012  032777  000001  164006  DRP:    BIT     #BIT0,@SWR      ;DROP DRIVE?
2647  015020  001464                          BEQ     RETT            ;NO
2648  015022  013704  001160                  MOV     UNNUM,R4        ;GET UNIT #
2649  015026  04!737  014032                  JSR     PC,GETERR       ;GET ERRORS
2650  015032  063737  001202  001206          ADD     WRITER,READER   ;ADD THE ERRORS
2651  015040  063737  001206  001204          ADD     READER,WCERR
2652  015046  063737  001204  001210          ADD     WCERR,COMERR
2653  015054  022737  000023  001210          CMP     #23,COMERR      ;DROP DRIVE?
2654  015062  103043                          BHIS    RETT            ;NO
2655  015064  053737  001164  001124          BIS     UNCMP,DROP      ;DROP DRIVE
2656  015072  104402  015076                  TYPE    .+2             ;.ASCIZ <15><12>"DROPPED UNIT "
2657  015116  013746  001160                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
2658  015122  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
2659  015124  104402  000636                  TYPE    ,CRLFLF
2660  015130  113703  001124                  MOVB    DROP,R3         ;GET DROPPED UNITS
2661  015134  113704  001162                  MOVB    UNITSV,R4       ;GET ALL DRIVES
2662  015140  020304                          CMP     R3,R4           ;ALL DRIVES DROPPED?
2663  015142  001003                          BNE     2$              ;NO
2664  015144  000000                          HALT                    ;NO MORE DRIVES
2665  015146  000137  001234                  JMP     @#BEGIN         ;RESTART TEST
2666  015152  032737  100000  001126  2$:     BIT     #BIT15,FLAG     ;DID OPERATOR SELECT PATTERN
2667  015160  001002                          BNE     3$              ;YES
2668  015162  005037  001136                  CLR     PATNU           ;NO CLEAR IT
2669  015166  000137  006072          3$:     JMP     @#EXTPPR        ;GET NEXT DRIVE
2670  015172  000207                  RETT:   RTS     PC
```

```
                                        ;ROUTINE TO RESTORE LOADER
015174  013705  015220          RLDR:   MOV     LDR1,R5         ;GET FIRST ADDRESS OF WHERE LOADER IS
                                                                ;TO BE RESTORED
015200  012704  017446                  MOV     #17446,R4       ;ADDRESS WHERE LOADER IS STORED
015204  012702  000155                  MOV     #155,R2         ;WORD COUNT
015210  012425                  1$:     MOV     (R4)+,(R5)+     ;RESTORE
015212  005302                          DEC     R2
015214  001375                          BNE     1$
015216  000000                          HALT                    ;DONE
015220  017446                  LDR1:   .WORD   17446           ;FIRST ADDRESS WHERE LOADERS ARE SAVED

        172100                  PARCSR=172100
        000114                  PARVEC=114
015222  012737  015314  000114  .MAMK:  MOV     #.PARSRV,@#PARVEC
015230  012737  000340  000116          MOV     #340,@#PARVEC+2 ;SET PRI LEVEL TO 7
015236  013746  000004                  MOV     @#4,-(SP)       ;SAVE CURRENT ERROR VECTOR
015242  013746  000006                  MOV     @#6,-(SP)       ;SAVE PRIOITY LEVEL
015246  012737  000006  000004          MOV     #6,@#4
015254  012737  000002  000006          MOV     #RTI,@#6
015262  012700  172100                  MOV     #PARCSR,R0      ;GET FIRST CSR ADDR
015266  012702  000001                  MOV     #1,R2
015272  012720  000001          1$:     MOV     #1,(R0)+        ;SET ACTION ENABLE IF AVAILABLE
015276  006302                          ASL     R2              ;SHIFT AVAILABILITY INDICATOR
015300  103374                          BCC     1$
015302  012637  000006                  MOV     (SP)+,@#6       ;RESTORE ERROR VECTOR PRIORITY
015306  012637  000004                  MOV     (SP)+,@#4       ;AND INTERRUPT VECTOR
015312  000207                          RTS     PC
                                        ;PARITY MEMORY TRAP

015314  104402  000736          .PARSRV:        TYPE    ,PAREER
015320  052737  004000  001122          BIS     #BIT11,FLAG2    ;SET ERROR FLAG
015326  032737  000010  001122          BIT     #BIT3,FLAG2     ;WERE WE COMPARING DURING ERROR?
015334  001422                          BEQ     13$             ;NO
015336  104402  015342                  TYPE    ,.+2            ;.ASCIZ " ADDR="
015352  005737  001212                  TST     MMAVA           ;IS MEM MGMT ON?
015356  001406                          BEQ     12$             ;NO
015360  013746  177776                  MOV     PS,-(6)         ;GET PS
015364  013746  001150                  MOV     SAVE,-(6)       ;GET VIRTUAL ADDR
015370  104412                          TYPEA                   ;CONVERT TO PHY AND TYPE
015372  000403                          BR      13$             ;CONT
015374  013746  001150          12$:    MOV     SAVE,-(6)       ;GET ADDR
015400  104406                          TYPES                   ;TYPE IT
015402  032777  100000  163416  13$:    BIT     #BIT15,@SWR     ;HALT ON ERROR?
015410  001401                          BEQ     1$              ;NO
015412  000000                          HALT                    ;YES
015414  012706  000500          1$:     MOV     #500,SP         ;RESET STACK
015420  000137  003246                  JMP     EXMFLG          ;RESTART TEST
```

```
                                .SBTTL          $TYPE - TTY TYPEOUT ROUTINE

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

015424  010446          .TYPE:  MOV     R4,-(6)         ;SAVE R4
015426  010546                  MOV     R5,-(6)         ;SAVE R5
015430  017605  000004          MOV     @4(6),R5        ;GET ADDRESS TO BE TYPED
015434  032705  177400          BIT     #177400,R5      ;IS IT A TYPEM?
015440  001002                  BNE     1$              ;NO
015442  016605  000004          MOV     4(6),R5         ;GET ADDRESS OF CHARACTEF
015446  105715          1$:     TSTB    (R5)            ;TERMINATOR?
015450  001423                  BEQ     2$              ;GET OUT IF SO
015452  122715  000012          CMPB    #12,(R5)        ;IS THE CHAR A LINE FEED
015456  001012                  BNE     4$              ;NO - GET OUT
015460  113704  001015          MOVB    FILCHR+1,R4     ;GET THE FILL COUNT
015464  113777  001014  163332  5$:     MOVB    FILCHR,@#TPB    ;TYPE A FILLER
015472  105777  163320          TSTB    @#TPS           ;DONE YET?
015476  100375                  BPL     .-4             ;NO - WAIT
015500  005304                  DEC     R4              ;DEC COUNT
015502  001370                  BNE     5$              ;LOOP UNTIL 0
015504  112577  163314          MOVB    (R5)+,@#TPB     ;LOAD AND TYPE THE CHARACTER
015510  105777  163302          TSTB    @#TPS           ;IS THE PRINTER READY
015514  100375                  BPL     .-4             ;WAIT UNTIL IT IS
015516  000753                  BR      1$              ;GET THE NEXT CHARACTER
015520  017646  000004          MOV     @4(6),-(6)      ;GET ADDRESS TO BE TYPED
015524  062766  000002  000006  ADD     #2,6(6)         ;ADD 2 TO THE ADDRESS
015532  022666  000004          CMP     (6)+,4(6)       ;IS IT .+2?
015536  001006                  BNE     3$              ;NO
015540  062705  000002          ADD     #2,R5           ;ADD 2 TO THE ADDRESS
015544  042705  000001          BIC     #1,R5           ;BACK UP TO AN EVEN BYTE
015550  010566  000004          MOV     R5,4(6)         ;RESTORE ADDRESS
015554  012605          3$:     MOV     (6)+,R5         ;RESTORE R5
015556  012604                  MOV     (6)+,R4         ;RESTORE R4
015560  000002                  RTI                     ;RETURN
```

```
                                .SBTTL          $SCOPE - SCOPE LOOP HANDLER

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

015562  104444          .SCOPE: KBDIN                   ;GO CHECK FOR ^G
015564  032777  040000  163234          BIT     #SW14,@SWR      ;LOOP ON TEST?
015572  001045                        BNE     .KIT            ;LOOP ON TEST IS SET
015574  000416                        BR      3$              ;SKIP - NOP FOR XOR TESTER
015576  013746  000004                MOV     @#4,-(6)        ;PUSH @#4 ON STACK
015602  012737  015622  000004        MOV     #4$,@#4         ;SET FOR TIMEOUT
015610  005737  177060                TST     @#177060        ;ERROR ON XOR?
015614  012637  000004                MOV     (6)+,@#4        ;POP STACK INTO @#4
015620  000426                        BR      .SVLAD          ;NO ERROR - GO TO NEXT TEST
015622  022626                4$:     CMP     (6)+,(6)+       ;CLEAR STACK
015624  012637  000004                MOV     (6)+,@#4        ;POP STACK INTO @#4
015630  000426                        BR      .KIT            ;ERROR - LOOP ON TEST
015632  032777  004000  163166  3$:   BIT     #SW11,@SWR      ;KILL ITERATIONS
015640  001012                        BNE     .SVLAD          ;YES - KILL ITERATIONS
015642  105737  001001                TSTB    ICNT+1          ;FIRST ONE?
015646  001404                        BEQ     2$              ;BRANCH IF FIRST
015650  123737  015734  001001        CMPB    TIMES,ICNT+1    ;DONE?
015656  003013                        BGT     .KIT            ;BRANCH IF NOT
015660  112737  000001  001001  2$:   MOVB    #1,ICNT+1       ;FIRST ITERATION
015666  105237  001000                INCB    ICNT            ;COUNT TEST NUMBERS
015672  011637  001010        .SVLAD: MOV     (6),LAD         ;SAVE LOOP ADDRESS
015676  013777  001000  163124        MOV     ICNT,@DISPLAY   ;DISPLAY TEST NO. AND ITERATION COUNT
015704  000002                        RTI                     ;RETURN

015706  105237  001001        .KIT:   INCB    ICNT+1          ;INC THE ITERATION COUNT
015712  013777  001000  163110  .OVER:  MOV     ICNT,@DISPLAY   ;SET UP DISPLAY
015720  005737  001010                TST     LAD             ;FIRST ONE?
015724  001760                        BEQ     .SVLAD          ;YES
015726  013716  001010                MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
015732  000002                        RTI                     ;FIXES PS

015734  000001          TIMES:  1                       ;RUN 1 TIMES
```

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

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

015736  005237  001002          .HLT:   INC     ERRORS          ;INC THE ERROR COUNT
015742  104442                          KBDIN                   ;GO CHECK FOR ^G
015744  032777  020000  163054          BIT     #SW13,@SWR      ;SKIP TYPEOUT IF SET
015752  001025                          BNE     2$              ;SKIP TYPEOUTS
015754  104402  015760                  TYPE    .+2             ;.ASCIZ <15><12>
015764  011637  001012                  MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
015770  162737  000002  001012          SUB     #2,HLTADR       ;FUDGE ADDRESS
015776  117737  163010  016060          MOVB    @HLTADR,.HLTCT  ;GET HLT ARGUEMENT
016004  013746  001012                  MOV     HLTADR,-(6)     ;PUT HLTADR ON STACK
016010  104404                          TYPEO                   ;TYPE STACK IN OCTAL
016012  104402  016016                  TYPE    .+2             ;.ASCIZ " "
016022  004737  014234                  JSR     PC,$SREG        ;GO TO USER ERROR ROUTINE
016026  005777  162774          2$:     TST     @SWR            ;HALT ON ERROR
016032  100001                          BPL     .+4             ;SKIP IF CONTINUE
016034  000000                          HALT                    ;HALT ON ERROR!
016036  032777  001000  162762          BIT     #SW9,@SWR       ;CHECK FOR INHIBIT LOOP ON ERROR
016044  001003                          BNE     3$              ;SKIP IF LOOP ON ERROR
016046  105037  001001                  CLRB    ICNT+1          ;CLEAR ITERATION COUNT
016052  000002                          RTI                     ;RETURN
016054  000137  015706          3$:     JMP     .KIT            ;LOOP ON TEST UNTIL NO ERRORS

016060  000000          .HLTCT: 0                               ;HLT ARGUMENT
```

```
                                        .SBTTL          $OCTAL - OCTAL TYPEOUT ROUTINE
                                        ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
                                        ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
                                        ;16 BITS. IT IS CALLED VIA THE TYPOCT, TYPBIT, OR TYPOCS MACRO'S.
016062  012737  170101  016250  .TYPEB: MOV     #170101,.PR     ;SET BIT FLAG AND 16. CHARACTER COUNT
016070  000411                  BR      .PTIT           ;NOW TYPE IT IN BIT FORM
016072  112737  000001  016250  .TYPEO: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
016100  000402                  BR      .+6             ;SKIP
016102  005037  016250          .TYPES: CLR     .PR             ;SUPRESS LEADING ZERO'S
016106  112737  177772  016251          MOVB    #-6,.PR+1       ;SET COUNT
016114                          .PTIT:
016114  010446                  MOV     R4,-(6)         ;PUSH R4 ON STACK
016116  010546                  MOV     R5,-(6)         ;PUSH R5 ON STACK
016120  016605  000010          MOV     10(6),R5        ;GET THE DATA
016124  012704  016252          MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
016130  105014                  CLRB    (4)             ;CLEAR FIRST BYTE
016132  000411                  BR      .PRF            ;ROTATE FIRST BIT
016134  105014          .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
016136  032737  000100  016250          BIT     #100,.PR        ;BIT TYPING MODE?
016144  001004                  BNE     .PRF            ;YES - SKIP 2 ROTATES
016146  006105                  ROL     R5              ;ROTATE BIT INTO C
016150  106114                  ROLB    (4)             ;PACK IT
016152  006105                  ROL     R5              ;ROTATE BIT INTO C
016154  106114                  ROLB    (4)             ;PACK IT
016156  006105          .PRF:   ROL     R5              ;ROTATE BIT INTO C
016160  106114                  ROLB    (4)             ;PACK IT
016162  105714                  TSTB    (4)             ;IS IT ZERO?
016164  001402                  BEQ     .+6             ;SKIP INC
016166  105237  016250          INCB    .PR             ;SET FILL SWITCH
016172  105737  016250          TSTB    .PR             ;CHECK FILL SWITCH
016176  001402                  BEQ     .+6             ;SKIP BITSET
016200  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
016204  105237  016251          INCB    .PR+1           ;INC COUNT
016210  001351                  BNE     .PRL            ;REPEAT
016212  022704  016252          CMP     #.PR+2,R4       ;EMPTY BUFFER?
016216  001002                  BNE     .+6             ;SKIP IF NOT
016220  112724  000060          MOVB    #'0,(4)+        ;LOAD 1 ZERO
016224  105014                  CLRB    (4)             ;NULL TERMINATOR
016226  104402  016252          TYPE    .PR+2           ;TYPE IT
016232  012605                  MOV     (6)+,R5         ;POP STACK INTO R5
016234  012604                  MOV     (6)+,R4         ;POP STACK INTO R4
016236  016666  000002  000004  MOV     2(6),4(6)       ;GET RID OF
016244  012616                  MOV     (6)+,(6)        ;DATA WORD
016246  000002                  RTI                     ;RETURN

016250  000012          .PR:    .BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
```

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

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

016274  012777  016422  000126  .POWER: MOV     #.ILLUP,@.PUVEC ;SET FOR FAST UP
016302  012777  000340  000122          MOV     #340,@.PUVEC$+2 ;PRIO:7
016310  010046                        MOV     R0,-(6)         ;PUSH R0 ON STACK
016312  010146                        MOV     R1,-(6)         ;PUSH R1 ON STACK
016314  010246                        MOV     R2,-(6)         ;PUSH R2 ON STACK
016316  010346                        MOV     R3,-(6)         ;PUSH R3 ON STACK
016320  010446                        MOV     R4,-(6)         ;PUSH R4 ON STACK
016322  010546                        MOV     R5,-(6)         ;PUSH R5 ON STACK
016324  010637  016426                MOV     SP,.SAVR6       ;SAVE SP
016330  012777  016340  000072        MOV     #.POWUP,@.PUVEC ;SET UP VECTOR
016336  000000                        HALT                    ;WAIT FOR PF

016340  013706  016426  .POWUP: MOV     .SAVR6,SP       ;GET SP
016344  005001                  CLR     R1              ;WAIT LOOP FOR THE TTY
016346  005201          1$:     INC     R1              ;WAIT FOR THE INC
016350  001376                  BNE     1$              ;OF  WORD
016352  012605                  MOV     (6)+,R5         ;POP STACK INTO R5
016354  012604                  MOV     (6)+,R4         ;POP STACK INTO R4
016356  012603                  MOV     (6)+,R3         ;POP STACK INTO R3
016360  012602                  MOV     (6)+,R2         ;POP STACK INTO R2
016362  012601                  MOV     (6)+,R1         ;POP STACK INTO R1
016364  012600                  MOV     (6)+,R0         ;POP STACK INTO R0
016366  012737  016274  000024  MOV     #.POWER,@#24    ;SET UP THE POWER DOWN VECTOR
016374  012737  000340  000026  MOV     #340,@#26       ;PRIO:7
016402  104402  016406          TYPE    ,.+2            ;.ASCIZ <15><12>"POWER"
016416  000137  007016          JMP     @TIMUP          ;JMP TO USER ADDRESS

016422  000000          .ILLUP: HALT                    ;THE POWER UP SEQUENCE WAS STARTED
016424  000776                  BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE

016426  000000          .SAVR6: 0                       ;PUT THE SP HERE
016430  000024  000026  .PUVEC: 24,26                   ;POWER UP VECTOR
```

```
                                        .SBTTL          $TYPEA - 18 BIT ADDRESS TYPER

                                        ;THIS ROUTINE TAKES 2 ARGUMENTS OFF THE STACK (OLD
                                        ;SP AND ADDRESS) AND, USING THE MEMORY MANAGEMENT REGISTERS, TYPES
                                        ;THE ADDRESS SUPPLIED IN 18 BIT FORM. THIS ROUTINE IS LINKED
                                        ;VIA THE 'TYPADR' MACRO.

016434                                  .TYPEA:
016434  010446                                  MOV     R4,-(6)         ;PUSH R4 ON STACK
016436  010546                                  MOV     R5,-(6)         ;PUSH R5 ON STACK
016440  016605  000012                          MOV     12(6),R5        ;R5 - OLD PS WITH PREVIOUS MODE
016444  016604  000010                          MOV     10(6),R4        ;R4 - ADDRESS TO BE DECODED AND TYPED
016450  016666  000006  000010                  MOV     6(6),10(6)      ;MOVE
016456  016666  000004  000006                  MOV     4(6),6(6)       ;DOWN
016464  016666  000002  000004                  MOV     2(6),4(6)       ;FOUR
016472  012616                                  MOV     (6)+,(6)        ;WORDS
016474  010346                                  MOV     R3,-(6)         ;PUSH R3 ON STACK
016476  000305                                  SWAB    R5              ;GET THE
016500  006005                                  ROR     R5              ;2 PREVIOUS
016502  006005                                  ROR     R5              ;MODE BITS
016504  006005                                  ROR     R5              ;INTO POSITION
016506  042705  177771                          BIC     #177771,R5      ;TO USE AS AN OFFSET
016512  016505  016632                          MOV     .SATAB(5),R5    ;R5 - SPACE ADDRESS FOR MM
016516  010403                                  MOV     R4,R3           ;R3 - REGISTER OFFSET
016520  042704  160000                          BIC     #160000,R4      ;CLEAR THE MM REG SELECT BITS
016524  000303                                  SWAB    R3              ;NOW MAKE
016526  006003                                  ROR     R3              ;MM REG
016530  006003                                  ROR     R3              ;SELECT BITS
016532  006003                                  ROR     R3              ;INTO AN
016534  006003                                  ROR     R3              ;OFFSET
016536  042703  177761                          BIC     #177761,R3      ;CLEAR THE JUNK BITS
016542  060305                                  ADD     R3,R5           ;ADD THE OFFSET TO THE TABLE
016544  011505                                  MOV     (5),R5          ;GET THE ISAR DATA
016546  006305                                  ASL     R5              ;THIS IS
016550  006305                                  ASL     R5              ;TO SHIFT
016552  006305                                  ASL     R5              ;THE SEGMENT
016554  006305                                  ASL     R5              ;ADDRESS
016556  005003                                  CLR     R3              ;INTO AN
016560  006305                                  ASL     R5              ;AN 18 BIT
016562  006103                                  ROL     R3              ;ADDRESS
016564  006305                                  ASL     R5              ;POSITION
016566  006103                                  ROL     R3              ;WITH R3 CONTAINING
016570  060405                                  ADD     R4,R5           ;THE UPPER 2 BITS
016572  005503                                  ADC     R3              ;AND R5 CONTAINING
016574  006305                                  ASL     R5              ;THE 16 BIT ADDRESS
016576  006103                                  ROL     R3              ;THEN SHIFT FOR TYPING
016600  110337  016252                          MOVB    R3,.PR+2        ;GET THE FIRST NUMBER FROM R3
016604  062737  000060  016252                  ADD     #'0,.PR+2       ;MAKE IT INTO A NUMBER
016612  012704  016253                          MOV     #.PR+3,R4       ;FUDGE IN THE POINTER
016616  012737  175401  016250                  MOV     #175401,..PR    ;AND THE FLAGS (FILL & 5 BYTES
016624  012603                                  MOV     (6)+,R3         ;POP STACK INTO R3
016626  000137  016134                          JMP     .PRL            ;DECODE AND TYPE THE REST

016632  172340                          .SATAB: 172340                  ;KISAR0
016634  172240                                  172240                  ;SISAR0
016636  172340                                  172340                  ;KISAR0 - NEVER USED
```

```
2971  016640  177640                    177640                    :UISARO
```

```
                                .SBTTL          $TRAP - TRAP HANDLER

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

016642  011646                  .TRAP:  MOV     (6),-(6)        ;GET ADDRESS OF TRAP +2
016644  162716  000002                  SUB     #2,(6)          ;MAKE IT ADDRESS OF TRAP
016650  017616  000000                  MOV     @(6),(6)        ;GET TRAP INSTRUCTION
016654  062716  112262                  ADD     #.TRP+2-TRAP,(6);GET DATA AND MAKE IT AN OFFSET
016660  01360?                  .TRP:   MOV     @(6)+,PC        ;GO TO PROPER SUBROUTINE

016662  015562                          .SCOPE                  ;SCOPE  =  TRAP+0       (104400)
016664  015424                          .TYPE                   ;TYPE   =  TRAP+2       (104402)
016666  016072                          .TYPEO                  ;TYPEO  =  TRAP+4        104404)
016670  016102                          .TYPES                  ;TYPES  =  TRAP+6       (104406)
016672  021364                          .TYPED                  ;TYPED  =  TRAP+10      (104410)
016674  016434                          .TYPEA                  ;TYPEA  =  TRAP+12      (104412)
016676  006124                          .ERCLR                  ;ERCLR  =  TRAP+14      (104414)
016700  006146                          .DKCMD                  ;DKCMD  =  TRAP+16      (104416)
016702  017304                          .RDOCT                  ;RDOCT  =  TRAP+20      (104420)
016704  016730                          .RDLIN                  ;RDLIN  =  TRAP+22      (104422)
016706  021216                          .UPDAT                  ;UPDAT  =  TRAP+24      (104424)
016710  000316                          .CLRDV                  ;CLRDV  =  TRAP+26      (104426)
016712  014126                          .LOGW                   ;LOGW   =  TRAP+30      (104430)
016714  014136                          .LOGR                   ;LOGR   =  TRAP+32      (104432)
016716  014144                          .LOGWC                  ;LOGWC  =  TRAP+34      (104434)
016720  014156                          .LOGC                   ;LOGC   =  TRAP+36      (104436)
016722  017214                          .CNTLU                  ;CNTLU  =  TRAP+40      (104440)
016724  017132                          .KBDIN                  ;KBDIN  =  TRAP+42      (104442)
016726  020000                          .SUSWP                  ;SUSWP  =  TRAP+44      (104444)
```

```
3004                                    .SBTTL          $RDLIN - TTY INPUT ROUTINE
3005
3006                                    ;THIS ROUTINE INPUTS A LINE TERMINATED BY A RETURN INTO ADDRESS
3007                                    ;INPUT AND RETURNS A LINE FEED. THE BUFFER HAS A NULL TERMINATOR
3008                                    ;INSTEAD OF THE RETURN. RUBOUTS ARE HANDLED BY RETYPING
3009                                    ;THE LINE. BUFFER OVERFLOW ERRORS LIKE A RUBOUT.
3010
3011  016730  010546                    .RDLIN: MOV     R5,-(6)         ;SAVE R5
3012  016732  012705  017052            1$:     MOV     #INPUT,R5       ;GET ADDRESS
3013  016736  022705  017072            2$:     CMP     #INPUT+16.,R5   ;BUFFER FULL?
3014  016742  001423                            BEQ     4$              ;YES - TYPE "?"
3015  016744  105737  177560                    TSTB    @#177560        ;WAIT FOR
3016  016750  100375                            BPL     .-4             ;A CHARACTER
3017  016752  113715  177562                    MOVB    @#177562,(5)    ;GET CHARACTER
3018  016756  142715  000200                    BICB    #200,(5)        ;GET RID OF JUNK
3019  016762  122715  000025                    CMPB    #25,(5)         ;IS IT A ↑U
3020  016766  001006                            BNE     5$              ;BRANCH IF NOT
3021  016770  104402  016774                    TYPE    ,.+2            ;.ASCIZ "↑U"<15><12>
3022  017002  000753                            BR      1$              ;START OVER
3023  017004  122715  000177            5$:     CMPB    #177,(5)        ;IS IT A RUBOUT
3024  017010  001005                            BNE     3$              ;SKIP IF NOT
3025  017012                            4$:
3026  017012  104402  017016                    TYPE    ,.+2            ;.ASCIZ "?"<15><12>
3027  017022  000743                            BR      1$              ;ZAP THE BUFFER AND LOOP
3028  017024  111527  000000            3$:     MOVB    (5),#0          ;SET UP FOR TYPING
3029  017030  104402  017026                    TYPE    ,3$+2           ;ECHO IT
3030  017034  122725  000015                    CMPB    #15,(5)+        ;CHECK FOR RETURN
3031  017040  001336                            BNE     2$              ;LOOP IF NOT RETURN
3032  017042  104402  000012                    TYPE    ,12             ;TYPE A LINE FEED
3033  017046  012605                            MOV     (6)+,R5         ;RESTORE R5
3034  017050  000002                            RTI                     ;RETURN
3035
3036  017052  000020                    INPUT:  .BLKB   16.             ;TTY INPUT AREA
3037  017072  000020                    ERTAB:  .BLKW   16.
3038
3039  017132  005737  000042                    .KBDIN: TST     42              ;GOT XXDP OR ACT
3040  017136  001057                            BNE     OKT             ;YES,GET OUT
3041  017140  022737  000176  001026            CMP     #SWREG,SWR      ;GOT SWITCH-LESS MACHINE?
3042  017146  001053                            BNE     OKT             ;NO GET OUT
3043  017150  105777  161644                    TSTB    @TKS            ;HAVE A CHARACTER
3044  017154  100050                            BPL     OKT             ;NO GET OUT
3045  017156  017737  161640  017300            MOV     @TKB,.MSG
3046  017164  042737  177600  017300            BIC     #177600,.MSG    ;STRIP OFF GARBAGE
3047  017172  122737  000007  017300            CMPB    #7,.MSG         ;DO WE HAVE A ↑G
3048  017200  001036                            BNE     OKT             ;NO,GET OUT
3049  017202  104402  017206                    TYPE    ,.+2            ;.ASCIZ <15><12>"↑G"
3050  017214                            .CNTLU:
3051  017214  104402  017220                    TYPE    ,.+2            ;.ASCIZ <15><12>"SWR= "
3052  017230  013746  000176                    MOV     $SWREG,-(6)     ;PUT SWREG ON STACK
3053  017234  104404                            TYPEO                   ;TYPE STACK IN OCTAL
3054  017236  104402  017242                    TYPE    ,.+2            ;.ASCIZ "    NEW= "
3055  017254  104420                            RDOCT
3056  017256  012637  017300                    MOV     (SP)+,.MSG      ;GET NEW VALUE OFF STACK
3057  017262  005737  017302                    TST     CTN             ;DID HE TYPE <CR> OF 000000?
3058  017266  001403                            BEQ     OKT             ;DONT CHANGE IF <CR>
3059  017270  013737  017300  000176            MOV     .MSG,SWREG      ;CHANGE VALUE OF SWREG
```

```
3060  017276  000002                  OKT:    RTI                     ;ALL DONE-EXIT
3061
3062  017300  000000                  .MSG:   0
3063  017302  000000                  CTN:    0
3064                                  .SBTTL          $RDOCT - OCTAL INPUT ROUTINE
3065
3066                                  ;THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
3067                                  ;IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.
3068
3069  017304  011646                  .RDOCT: MOV     (6),-(6)        ;MOVE THE PC
3070  017306  016666  000004  000002          MOV     4(6),2(6)       ;MOVE THE PS
3071  017314  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
3072  017316  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
3073  017320  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
3074  017322  104422                  4$:     RDLIN                   ;READ A LINE INTO INPUT
3075  017324  005001                          CLR     R1              ;INIT DATA WORD
3076  017326  005037  017302                  CLR     CTN             ;CLEAR COUNT WORD
3077  017332  012703  017052                  MOV     #INPUT,R3       ;INIT POINTER
3078  017336  112302                  1$:     MOVB    (3)+,R2         ;GET A BYTE
3079  017340  122702  000015                  CMPB    #15,R2          ;WAS IT A <CR>?
3080  017344  001421                          BEQ     2$              ;GET OUT IF YES
3081  017346  122702  000060                  CMPB    #'0,R2          ;CHECK FOR 0 OR GREATER
3082  017352  003024                          BGT     3$              ;ERROR - LESS THAN 0
3083  017354  122702  000067                  CMPB    #'7,R2          ;CHECK FOR 7 OR LESS
3084  017360  002421                          BLT     3$              ;ERROR - GREATER THAN 7
3085  017362  006002                          ROR     R2              ;GET
3086  017364  006002                          ROR     R2              ;INTO
3087  017366  006002                          ROR     R2              ;POSITION
3088  017370  006101                          ROL     R1              ;FIRST BIT
3089  017372  006102                          ROL     R2              ;GET
3090  017374  006101                          ROL     R1              ;SECOND BIT
3091  017376  006102                          ROL     R2              ;GET
3092  017400  006101                          ROL     R1              ;THIRD BIT
3093  017402  005237  017302                  INC     CTN             ;YES HE TYPED SOMETHING
3094  017406  000753                          BR      1$              ;LOOP
3095  017410  010166  000012          2$:     MOV     R1,12(6)        ;SAVE THE RESULT
3096  017414  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
3097  017416  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
3098  017420  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
3099  017422  000002                          RTI                     ;RETURN
3100
3101  017424                          3$:
3102  017424  104402  017430                  TYPE    ,.+2            ;.ASCIZ "?"<15><12>
3103  017434  000732                          BR      4$              ;TRY AGAIN
3104  017436  000000                  OUTBUF: 0
3105
3106                                  ;NOTE FOR PROGRAMMER******** PROGRAM AT THIS POINT CAN NOT EXCEED A PC OF 17444*****
```

```
                020000                          .=      20000
                                                ;NOTE    ALL THIS CODE GETS DESTROYED WHEN PATTERN IS WRITTEN
020000  032737  000001  020116  .SUSWR: BIT     #BIT0,SWI
020006  001037                          BNE     XXX
020010  013746  000006                  MOV     6,-(SP)         ;SAVE 6 ON STACK
020014  013746  000004                  MOV     4,-(SP)         ;SAVE 4 ON STACK
020020  012737  020040  000004          MOV     #1$,4           ;SET UP TRAP ADDRESS
020026  022777  177777  160772          CMP     #-1,@SWR        ;TEST 177570
020034  001402                          BEQ     2$              ;FAKE OUT
020036  000407                          BR      3$              ;HARDWARE AVAILABLE
020040  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
020042  012737  000176  001026  2$:     MOV     #SWREG,SWR      ;SET UP SOFTWARE REGISTERS
020050  012737  000174  001030          MOV     #DISPREG,DISPLAY
020056  022737  000176  001026  3$:     CMP     #SWREG,SWR      ;1ST TIME THRU?
020064  001004                          BNE     4$              ;NO CHANGE STILL 177570
020066  005737  000042                  TST     42              ;ANY XXDP OR ACT
020072  001001                          BNE     4$              ;SWR=000000
020074  104440                          CNTLU                   ;GET INITIAL SETTINGS
020076  012637  000004          4$:     MOV     (SP)+,4         ;REPLACE 4 FROM STACK
020102  012637  000006                  MOV     (SP)+,6         ;REPLACE 6 FROM STACK
020106  052737  000001  020116  XXX:    BIS     #BIT0,SWI       ;SET THE BEENHEREBIT
020114  000002                          RTI                     ;ALL DONE

020116  000000                  SWI:    0


                                        ;ROUTINE TO SAVE ABS LOADER
020120  012700  017776          LDR:    MOV     #17776,R0
020124  012737  020144  000004          MOV     #2$,4           ;SET TIME OUT TRAP VECTOR
020132  012737  000340  000006          MOV     #340,6
020140  005720                          TST     (R0)+
020142  000776                          BR      .-2
020144  022626                  2$:     CMP     (SP)+,(SP)+
020146  012737  000006  000004          MOV     #6,4
020154  005037  000006                  CLR     6
020160  162700  000334                  SUB     #334,R0         ;POINT R0 BACK TO LOADER
020164  010037  015220                  MOV     R0,LDR1         ;SAVE FOR RESTORE ROUTINE
020170  012702  000155                  MOV     #155,R2         ;WORD COUNT
020174  012703  017446                  MOV     #17446,R3       ;WHERE LOADER IS TO BE STORED
020200  012023                  1$:     MOV     (R0)+,(R3)+     ;STORE LOADER
020202  005302                          DEC     R2
020204  001375                          BNE     1$
020206  000207                          RTS     PC              ;RETURN


                                        ; -A- PORT SIZE

020210  052737  020000  001122  SIZZAP: BIS     #BIT13,FLAG2    ;SET MAPPING BIT
020216  042737  000500  001126          BIC     #500,   FLAG    ;CLEAR FLAG BEFORE READ
020224  042737  000460  001122          BIC     #460,   FLAG2   ;CLEAR FLAG2 BEFORE READ
020232  042737  003400  001172          BIC     #3400,  CMD     ;CLEAR CMD BEFORE READ
020240  004737  001702                  JSR     PC,DRVENO       ;FIND DRIVE
020244  012737  000002  001224          MOV     #2,WORK1        ;START WITH ONE 4K BUFFER
020252  012737  000001  001070          MOV     #1,STAMEM       ;FIRST 4K BANK
```

```
 3163  020260  012737  057476  001140          MOV     #57476,BUF      ;GET STARTING ADDR. 5K
 3164  020266  012737  000001  001130          MOV     #1,WRDCT        ;LOAD WC
 3165  020274  005037  001134                  CLR     DMA             ;LOAD DA
 3166  020300  012777  000040  160526          MOV     #40,@RSCS2      ;CLEAR ALL RS REG
 3167  020306  013777  001160  160520          MOV     UNNUM,@RSCS2    ;GET DRIVE #
 3168  020314  012737  000071  001172          MOV     #71,CMD         ;DO A READ
 3169  020322  104416                  4$:     DKCMD                   ;NOW
 3170  020324  105777  160502          1$:     TSTB    @RSCS1          ;DONE YET?
 3171  020330  100375                          BPL     1$              ;NO
 3172  020332  032777  004000  160474          BIT     #4000,@RSCS2    ;DID NEM SET?
 3173  020340  001012                          BNE     SIZ1            ;YES
 3174  020342  005777  160464                  TST     @RSCS1          ;ANY ERRORS?
 3175  020346  100005                          BPL     3$              ;NO
 3176  020350  012737  000006  001106          MOV     #6,SIZEAP       ;GET SIZE OF BUFFER
 3177  020356  000137  020572                  JMP     @#SIZERR        ;FOR USER IF HE WISHES IT
 3178  020362  104424                  3$:     UPDAT                   ;GET NEXT 4K BANK
 3179  020364  000756                          BR      4$              ;TEST BANK
 3180  020366  005337  001224          SIZ1:   DEC     WORK1           ;DEC SIZE OF BUFFER
 3181  020372  013737  001224  001106          MOV     WORK1,SIZEAP    ;LOAD SIZE OF A BUFFER
 3182  020400  104402  020404                  TYPE    .+2             ;.ASCIZ <15><12>"PORT -A- DATA BUFFER     4K TO "
 3183  020446  004737  021156                  JSR     PC,SIZPR
```

```
                                        ; -B- PORT SIZE
                                        
003184                                  
003185                                  
003186  020452  012737  000001  001224  SIZZBP: MOV     #1,WORK1        ;START WITH ONE 4K BUFFER
003187  020460  042737  000500  001126          BIC     #500,   FLAG    ;CLEAR FLAG BEFORE READ
003188  020466  042737  000460  001122          BIC     #460,   FLAG2   ;CLEAR FLAG2 BEFORE READ
003189  020474  042737  003400  001172          BIC     #3400,  CMD     ;CLEAR CMD BEFORE READ
003190  020502  012737  037476  001140          MOV     #37476,BUF      ;GET STARTING ADDR. 4K
003191  020510  012737  000001  001130          MOV     #1,WRDCT        ;LOAD WC
003192  020516  005037  001134                  CLR     DMA             ;LOAD DA
003193  020522  012777  000040  160304          MOV     #40,@RSCS2      ;CLEAR ALL RS REG
003194  020530  013777  001160  160276          MOV     UNNUM,@RSCS2    ;GET DRIVE #
003195  020536  012737  002071  001172          MOV     #2071,CMD       ;DO A READ
003196  020544  104416                  SIZ2:   DKCMD                   ;NOW
003197  020546  105777  160260          1$:     TSTB    @RSCS1          ;DONE YET?
003198  020552  100375                          BPL     1$              ;NO
003199  020554  032777  004000  160252          BIT     #4000,@RSCS2    ;DID NEM SET?
003200  020562  001067                          BNE     SIZB1           ;YES
003201  020564  005777  160242                  TST     @RSCS1          ;ANY ERRORS?
003202  020570  100047                          BPL     SIZ3            ;NO
003203  020572                          SIZERR:
003204  020572  104402  020576                  TYPE    .+2             ;.ASCIZ <15><12>"WILL NOT CONTINUE TO SIZE MEMORY BECAUS
003205  020662  012737  000006  001110          MOV     #6,SIZEBP       ;GIVE PROGRAM A BUFFER
003206  020670  104060                          HLT     !BA!DS          ;YOU CAN ENTER CONVERSATION MODE
003207  020672  052737  000001  001122          BIS     #BIT0,FLAG2     ;BEEN HERE BEFORE FLAG
003208  020700  042737  020000  001122          BIC     #BIT13,FLAG2    ;CLEAR MAPPING FLAG
003209  020706  000000                          HALT                    ;OR GO TO DZRSB
003210  020710  032737  000400  001122  SIZ3:   BIT     #BIT8,FLAG2     ;FOUND MEMORY YET?
003211  020716  001006                          BNE     SIZB3           ;YES
003212  020720  052737  000400  001122          BIS     #BIT8,FLAG2     ;SET FOUND MEM FLAG
003213  020726  013737  001224  001074          MOV     WORK1,STBCOM    ;GET 1ST BANK
```

```
3214  020734  104424                    SIZB3:  UPDAT                   ;GET NEXT 4K BANK
3215  020736  000702                            BR      SIZ2            ;TEST BANK
3216  020740  000404                            BR      SIZB2
3217  020742  032737  000400  001122    SIZB1:  BIT     #BIT8,FLAG2     ;FOUND MEMORY?
3218  020750  001771                            BEQ     SIZB3           ;NO
3219  020752  005337  001224            SIZB2:  DEC     WORK1           ;DEC SIZE OF BUFFER
3220  020756  013737  001224  001110            MOV     WORK1,SIZEBP    ;LOAD SIZE OF B BUFFER
3221  020764  032737  000400  001122            BIT     #BIT8,FLAG2     ;FOUND MEMORY?
3222  020772  001017                            BNE     6$              ;YES
3223  020774  104402  021000                    TYPE    .+2             ;.ASCIZ <15><12>"NO MEMORY ON -B- PORT"
3224  021030  000442                            BR      1$
3225  021032                            6$:
3226  021032  104402  021036                    TYPE    .+2             ;.ASCIZ <15><12> "PORT -B- DATA BUFFER"
3227  021066  013737  001074  001224            MOV     $TBCOM,WORK1
3228  021074  005001                            CLR     R1
3229  021076  005002                            CLR     R2
3230  021100  004737  021164                    JSR     PC,SIZP
3231  021104  104402  021110                    TYPE    .+2             ;.ASCIZ " TO "
3232  021116  013737  001110  001224            MOV     SIZEBP,WORK1
3233  021124  004737  021156                    JSR     PC,SIZPR
3234  021130  052737  000100  001126            BIS     #BIT6,FLAG      ;SET MULTI PORT FLAG
3235  021136  042737  020000  001122    1$:     BIC     #BIT13,FLAG2    ;CLEAR MAPPING FLAG
3236  021144  052737  000002  001122            BIS     #BIT1,FLAG2     ;SET BEEN HERE FLAG
3237  021152  000137  001552                    JMP     CALM            ;CAL BUFFER AND WC
3238
3239  021156  005001                    SIZPR:  CLR     R1              ;INIT SETUP
3240  021160  012702  000004                    MOV     #4,R2
3241  021164  062701  000001            SIZP:   ADD     #1,R1           ;SETUP FOR BANK NO
3242  021170  062702  000004                    ADD     #4,R2           ;SETUP FOR SIZE FO MEMORY
3243  021174  020137  001224                    CMP     R1,WORK1        ;IS THIS THE RIGHT SIZE?
3244  021200  001371                            BNE     SIZP            ;NO
3245  021202  010246                            MOV     R2,-(6)         ;PUT R2 ON STACK
3246  021204  104410                            TYPED                   ;TYPE STACK IN DECIMAL
3247  021206  104402  021212                    TYPE    .+2             ;.ASCIZ "K"
3248  021214  000207                            RTS     PC              ;RETURN
3249
3250                                    ;ADD 4K TO TEST ADDR
3251
3252  021216  005237  001224            .UPDAT: INC     WORK1           ;INC BANK #
3253  021222  062737  020000  001140            ADD     #20000,BUF      ;UPDATE BY 4K
3254  021230  005737  001212                    TST     MMAVA           ;MEMORY MANAGEMENT AVAILABLE?
3255  021234  001437                            BEQ     3$              ;NO
3256  021236  022737  000037  001224            CMP     #37,    WORK1   ;BANK 37?
3257  021244  001440                            BEQ     1$              ;YES-STOP SIZING
3258  021246  022737  000027  001224            CMP     #27,    WORK1   ;BANK 30?
3259  021254  002004                            BGE     10$             ;NO
3260  021256  052737  001400  001172            BIS     #1400,  CMD     ;YES-ENABLE A16 AND A17
3261  021264  000002                            RTI                     ;CONTINUE
3262  021266  022737  000017  001224    10$:    CMP     #17,    WORK1   ;BANK 20?
3263  021274  002007                            BGE     11$             ;NO
3264  021276  052737  001000  001172            BIS     #1000,  CMD     ;YES-ENABLE A17
3265  021304  042737  000400  001172            BIC     #400,   CMD     ;DISABLE A16
3266  021312  000002                            RTI                     ;CONTINUE
3267  021314  022737  000007  001224    11$:    CMP     #7,     WORK1   ;BANK 10?
3268  021322  002010                            BGE     2$              ;NO
3269  021324  052737  000400  001172            BIS     #400,   CMD     ;YES-ENABLE A16
```

```
                                        RTI                             ;CONTINUE
3271  021334  022737  177476  001140  3$:   CMP     #177476,BUF       ;EXCEEDED MEM YET?
3272  021342  001401                        BEQ     1$                ;YES
3273  021344  000002                  2$:   RTI                       ;CONTINUE
3274  021346  062716  000002        1$:   ADD     #2,(6)            ;UPDATE RETURN PC BY 2
3275  021352  042737  003400  001172        BIC     #3400,  CMD       ;CLEAR CMD AFTER FINISHING
3276  021360  000002                        RTI
3277
3278  021362  021364                        .TYPEC                          ;TYPED   =  TRAP+46      (104446)
3279                                  .SBTTL  $TYPEC - CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
3280
3281  021364  012737  100040  021612  .TYPED: MOV     #100040,.DSIGN    ;SET BLANK SWITCH AND SIGN
3282  021372  010046                        MOV     R0,-(6)           ;PUSH R0 ON STACK
3283  021374  010146                        MOV     R1,-(6)           ;PUSH R1 ON STACK
3284  021376  010246                        MOV     R2,-(6)           ;PUSH R2 ON STACK
3285  021400  010346                        MOV     R3,-(6)           ;PUSH R3 ON STACK
3286  021402  010546                        MOV     R5,-(6)           ;PUSH R5 ON STACK
3287  021404  012737  100040  021612        MOV     #100040,.DSIGN    ;SET BLANK SWITCH AND SIGN
3288  021412  016605  000016                MOV     16(6),R5          ;GET DATA TO BE TYPED
3289  021416  100004                        BPL     1$                ;BR IF INPUT IS POS.
3290  021420  005405                        NEG     R5                ;MAKE THE BINARY NUMBER POS.
3291  021422  112737  000055  021612        MOVB    #'-,.DSIGN        ;MAKE THE ASCII NUMBER NEG.
3292  021430  005000                  1$:   CLR     R0                ;ZERO THE CONSTANTS INDEX
3293  021432  012703  021602                MOV     #.DBLK,R3         ;SETUP THE OUTPUT POINTER
3294  021436  112723  000040                MOVB    #' ,(R3)+         ;SET THE FIRST CHARACTER TO A BLANK
3295  021442  005002                  2$:   CLR     R2                ;CLEAR THE BCD NUMBER
3296  021444  016001  021572                MOV     .DTBL(R0),R1      ;GET THE CONSTANT
3297  021450  160105                  3$:   SUB     R1,R5             ;FORM THIS BCD DIGIT
3298  021452  002402                        BLT     4$                ;BR IF DONE
3299  021454  005202                        INC     R2                ;INCREASE THE BCD DIGIT BY 1
3300  021456  000774                        BR      3$
3301  021460  060105                  4$:   ADD     R1,R5             ;ADD BACK THE CONSTANT
3302  021462  005702                        TST     R2                ;CHECK IF BCD DIGIT=0
3303  021464  001003                        BNE     5$                ;FALL THROUGH IF 0
3304  021466  105737  021613                TSTB    .DSIGN+1          ;STILL DOING LEADING 0'S?
3305  021472  100410                        BMI     7$                ;BR IF YES
3306  021474  106337  021613          5$:   ASLB    .DSIGN+1          ;MSD?
3307  021500  103003                        BCC     6$                ;BR IF NO
3308  021502  113763  021612  177777        MOVB    .DSIGN,-1(R3)     ;YES--SET THE SIGN
3309  021510  052702  000060          6$:   BIS     #'0,R2            ;MAKE THE BCD DIGIT ASCII
3310  021514  052702  000040          7$:   BIS     #' ,R2            ;MAKE IT A SPACE IF NOT ALREADY A DIGIT
3311  021520  110223                        MOVB    R2,(R3)+          ;PUT THIS CHARACTER IN THE OUTPUT BUFFER
3312  021522  005720                        TST     (R0)+             ;JUST INCREMENTING
3313  021524  020027  000010                CMP     R0,#10            ;CHECK THE TABLE INDEX
3314  021530  002744                        BLT     2$                ;GO DO THE NEXT DIGIT
3315  021532  003002                        BGT     8$                ;GO TO EXIT
3316  021534  010502                        MOV     R5,R2             ;GET THE LSD
3317  021536  000764                        BR      6$                ;GO CHANGE TO ASCII
3318  021540  105013                  8$:   CLRB    (R3)              ;SET THE TERMINATOR
3319  021542  012605                        MOV     (6)+,R5           ;POP STACK INTO R5
3320  021544  012603                        MOV     (6)+,R3           ;POP STACK INTO R3
3321  021546  012602                        MOV     (6)+,R2           ;POP STACK INTO R2
3322  021550  012601                        MOV     (6)+,R1           ;POP STACK INTO R1
3323  021552  012600                        MOV     (6)+,R0           ;POP STACK INTO R0
3324  021554  016666  000002  000004        MOV     2(6),4(6)         ;FUDGE DATA
3325  021562  012616                        MOV     (6)+,(6)          ;OFF STACK
```

```
  3326  021564  104402  021602                    TYPE    ,,DBLK          ;NOW TYPE THE NUMBER
  3327  021570  000002                            RTI                     ;RETURN
  3328
  3329  021572  023420  001750  000144  .DTBL:   10000.,1000.,100.,10.
  3330  021600  000012
  3331  021602  000004                  .DBLK:   .BLKW   4
  3332  021612  000000                  .DSIGN:  0
  3333
  3334          000001                           .END
```

```
ADAM    010702          2021    2026#   2035
ADDRR   014164          2542    2549    2554#
ADERR   003632          1236    1241#
ADHGT   003570          1225    1230#
ADRD    004736          1389    1406#
ADRI    004752          1405    1408    1410#
ADTL    003324           619    1172#
ADTST   003232           611     966    1154    1157#   1745    2536
ADTI    003332          1165    1182#
ADTIA   003334          1187#   1258
AKH     007356          1802#   1821
APORT   011354          1271    2107#
AS      = 000100         810#   1647    2591
ASKWC   002556          1065    1083#
AOBI    001114           830#    935#   1046#   1095    1103    1269    1423#   1628    1712    1809    1969    2450#   2509#
A1      002406           615    1058#
BA      = 000020         808#   1214    1355    1514    2596    3206
BEGIN   001234           910#   1980    1999    2287    2665
BELL    = 000007         697#
BIT0    = 000001         706#    922     928     946     951     955    2646    3110    3128    3207
BIT1    = 000002         707#    959     972    1026    1037    1081    1419    3236
BIT10   = 002000         716#    961     962    1058    1164    1421    1630    1661    2261    2288
BIT11   = 004000         717#   1064    1157    1581    1658    2701
BIT12   = 010000         718#   1149    1371    1536    2097    2444    2638
BIT13   = 020000         719#   1155    1313    1602    2566    3156    3208    3235
BIT14   = 040000         595     720#   1004    1143    1283    1684    1686
BIT15   = 100000         721#   1134    1172    1256    1428    1568    1977    2353    2361    2410    2420    2666    2713
BIT2    = 000004         708#    936     986    1011    1018    1160    1163    1187    1385    1438    1604    1755    1881
                        2095    2443    2568
BIT3    = 000010         709#   1707    1987    2026    2104    2467    2702
BIT4    = 000020         710#   1623    1710    2090    2226    2229
BIT5    = 000040         711#    983    1009    1110    1620    1670    1750    2223    2225    2510
BIT6    = 000100         712#   1055    1267    1325    1390    1472    1500    1543    1626    1807    2030    2144    215?
                        2174    3234
BIT7    = 000200         713#    816    1007    2197    2373    2432
BIT8    = 000400         714#   1161    1192    1255    1338    1441    1567    1616    1625    1705    1803    3210    3212
                        3217    3221
BIT9    = 001000         715#   1212    1226    1289    1320    1327    1388    1470    1498    1540    1545    1590    1593
                        1622    1637    1657    1708    1781    2039    2334    2350    2393    2407    2415    2421    2431
                        2459    2643
BLOCK   001154           884#   1774    1778#   1841#   1843#   1846#   1848    1850    1851#   1852    1865#   2057#   2071#
                        2083
BLSZ    007562          1773    1839#
BPORT   011422          1273    2117#
BUF     001140           878#   1198#   1277#   1316#   1464#   1612    2451#   3163#   3190#   3253#   3271
BUFEXI  007346          1752    1760    1766    1770    1777    1800#
BUFINX  007432          1799    1813#
CALM    001552           947     949#   3237
CALM1   001556           950#
CHCKDV  002226          1021    1026#
CHKADT  003656          1240    1246    1251#
CHKDAT  012646          2345#   2357
CHKDSK  013112          2403#   2414
CLEAR   011306          1369    1531    2091    2093    2095#
CLIND   014132          2541#   2545
CLRDV   = 104426        1195    1265    1448    1580    1592    2326    2344    2395    2393    2996#
```

```
CMD     001172          891# 1205# 1220# 1285# 1317# 1380# 1467# 1495# 1533# 1622# 1625# 1630# 1631
                       2331# 2346# 2404# 2453# 2456# 3159# 3168# 3189# 3195# 3260# 3264# 3265# 3269#
                       3275#
CMDAE   007340         1785  1793  1798#
CMPLP   010572         2005# 2006  2024
CMPLP1  010636         2002  2004  2016#
CMPY    003306         1044  1062  1085  1108  1141  1147  1153  1168#
CNTLU = 104440         3001# 3125
COMDAR  007244         1779  1781#
COMERR  001210          898# 2503  2523# 2652# 2653
COMPAR  010452         1387  1538  1986#
CONM    002276          967  1042#
CRLFLF  000636          750# 1013  1247  1526  2209  2659
CTN     017302         3057  3063# 3076# 3093#
DA    = 000004          806# 1214  1228  1297  1355  1399  1482  1514  1553  1998  2286  2336  2352
                       2461  2601
DATA    000554          734# 1295  1333  1396
DATAT   003716         1166  1189  1259# 1417
DATTES  002466         1063  1066# 1070
DB    = 000002          805# 1355  1514  1553  2336  2352  2409  2461  2616
DISBUF  007054         1216  1252  1308  1367  1410  1750# 2339  2356  2413  2463
DISPLA  001030          785# 1876# 2786# 2790# 3120#
DISPRE  000174          598# 3120
DKCMD = 104416         1206  1221  1286  1318  1381  1468  1496  1534  2332  2347  2405  2454  2457
                       2992# 3169  3196
DKINT   006360         1637# 1730
DMA     001134          876#  956# 1194# 1202  1235  1241  1264# 1457# 1458# 1579# 1600  1672# 1759
                       1761# 1765  1767# 1769  1771# 1775# 1776  1784  1789  1800# 1814  2082  2328#
                       2343# 2400# 2445# 3165# 3192#
DONE    014064         2511  2516  2528#
DONEE   002262         1027  1036# 1585
DOWN    012770         2329  2367# 2370
DROP    001124          834#  980  2301# 2471  2507  2655# 2660
DRP     015012         2563  2644  2646#
DRVENO  001702          971# 1159  3160
DS    = 000040          809# 1297  1399  1482  1647  1998  2286  2611  3206
DSKRD   004576         1377  1379# 1403
DT    = 000240          813# 2623
DVNUM   001742          976   979#
ELH     004640         1384  1386  1388#
ERCLR = 104414         1199  1219  1281  1311  1379  1466  1494  1532  2991#
ERCOUN  001146          881# 1986# 2092#
ERRCL   012452          933  1059  2294#
ERRORS  001002          775# 2805# 2826
ERTAB   017072         2295  2519  2560# 3037#
ESH     004532         1370# 1427
ESH1    004526         1315  1369#
EXGEN   010204         1920  1925#
EXGEN1  010350         1961#
EXMFLG  003246         1158  1160# 1584  2717
EXRAX   006014         1488  1520  1541  1559#
EXRXX   006026         1560  1562#
EXTDR   007560         1806  1812  1837#
EXTMEM  011476          949  1056  2135#
EXTPP   006120         1582  1585#
EXTPPR  006072         1424  1440  1579#  2669
```

```
EXIT    011420          2110    2115*   2121
FILCHR  001014           779*   2738    2739
FILDAT  010020          1886*   1888
FLAG    001126           873*    925*    961*    964*   1055*   1058*   1060*   1064*   1110*   1134*   1143*   1149*   1155*
                        1157    1160*   1163*   1164    1187    1192*   1212*   1226*   1255*   1267    1278*   1283    1289
                        1298*   1299    1307*   1313    1320    1371    1375*   1385    1388    1412    1421    1438    1441*
                        1459*   1470    1484*   1485    1496    1528*   1540    1567*   1581    1590    1593*   1604    1616
                        1626    1637*   1657*   1670    1750    1755    1781    1798    1802*   1803    1807    1836*   1881
                        2095    2097    2144    2154    2174    2334    2350    2407    2443*   2459    2666    3157*   3187*
                        3234*
FLAG2   001122           833*    926*    928     946     951     955*    972*    986*   1004    1011    1018*   1026    1037*
                        1081*   1602    1620    1623    1661    1684*   1686*   1987*   2026*   2104*   2223*   2225*   2226*
                        2229*   2261*   2288*   2353*   2358    2361*   2393*   2410*   2415    2417    2420*   2421*   2431*
                        2444*   2568*   2638    2701*   2702    3156*   3158*   3188*   3207*   3208*   3210    3212*   3217
                        3221    3235*   3236*
FLAG3   001216           901*    927*   1172*   1256    1428*   1568
FNDTYP  006544          1072    1684*   2325
GETERR  014032          2485    2517*   2649
HINUM   010212          1895    1897    1907    1921*   1927*
HINUM1  010354          1939    1949    1960*   1963*   1992*   2014
HISAV   010216          1895*   1929*   1992
HLT   = 104000           694*   1214    1228    1243    1297    1355    1399    1482    1514    1553    1647    1791    1993
                        2045    2286    2336    2352    2409    2461    3206
HLTADR  001012           778*   2810*   2811*   2812    2813    2827
HRDER   001152           883*   1276*   1450*   1655    2342*   2394*
ICNT    001000           774*    917*   2779    2781    2783*   2784*   2786    2789*   2790    2796    2822*
INCSEC  007220          1775*   1780
INPUT   017052          1170    3012    3013    3036*   3077
INTEXT  006450          1640    1658*
INTFLG  001176           893*   1601*   1654*   2000    2005    2028
KBDIN = 104442          2765    2806    3002*
KDPAR0= 172360           684*
KDPAR1= 172362           685*
KDPAR2= 172364           686*
KDPAR3= 172366           687*
KDPAR4= 172370           688*
KDPAR5= 172372           689*
KDPAR6= 172374           690*
KDPAR7= 172376           691*
KDPDR0= 172320           666*
KDPDR1= 172322           667*
KDPDR2= 172324           668*
KDPDR3= 172326           669*
KDPDR4= 172330           670*
KDPDR5= 172332           671*
KDPDR6= 172334           672*
KDPDR7= 172336           673*
KIPAR0= 172340           675*    2215*
KIPAR1= 172342           676*    2217*
KIPAR2= 172344           677*    2233*
KIPAR3= 172346           678*    2235*
KIPAR4= 172350           679*    2237*
KIPAR5= 172352           680*    2239*
KIPAR6= 172354           681*    2241*
KIPAR7= 172356           682*    2250*
KIPDP0= 172300           657*    2242*
```

```
KIPDR1= 172302          658#   2243*
KIPDR2= 172304          659#   2244*
KIPDR3= 172306          660#   2245*
KIPDR4= 172310          661#   2246*
KIPDR5= 172312          662#   2247*
KIPDR6= 172314          663#   2248*
KIPDR7= 172316          664#   2249*
LA    = 000204          811#   2633
LAD     001010          777#    918*  1211*  1288*  1366*  1453*  2785*  2791   2793   2796
LDAT    004030         1276    1281#  1288   1300
LDAT1   004024         1280#   1309
LDR     020120          930    3136#
LDR1    015220         2672    2680#  3145*
LOADSW  000510          725#   1066   2321   2482
LOGC  = 104436         1244    2038   3000#
LOGR  = 104432         1227    1398   1542   1786   2998#
LOGW  = 104430         1213    1291   1476   2997#
LOGWC = 104434         1324    1504   2999#
LONUM   010210         1894    1896   1905   1917*  1926#
LONUM1  010352         1938    1947   1959*  1962#  1991*  2010
LOPCNT  001200          894#   1280*  1292*  1302   1305*  1322   1329   1357*  1358   1361   1364*  1376*  1392
                       1401*   1402   1407   1465*  1474   1477*  1489   1493*  1502   1515*  1516   1519*  1521
                       1524    1530*  1547   1555*  1556   1559   1681*  2032   2129*  2207   2554
LOSAV   010214         1894*   1928#  1991
MDON    012324         2214    2254#
MEM     010356         1870    1967#
MMABTO  012326         2252    2257#
MMAVA   001212          899#    934*   941*  1343   1444   1618   2049   2109   2120   2139   2212   2705   3254
MMSET   011402         2112#   2123
MMVEC = 000250          819#   2252*
MR    = 000220          812#   2628
MSTR    004760         1373    1412#
MYBYWR  012562         2331#   2340
N     = 000005          693#   1183   1187#  1260   1264#  1434   1438#  1575   1579#
NOPORT  002374         1051    1056#
NOREC   006524         1360    1404   1558   1678#
NOWGO   002266         1019    1035   1037#
NPRCNT  012446         2262*   2284*  2290#
NPRRET  012436         1663    2288#
NPR1    012450         2263    2291#
NUMS    001170          890#   1068*  1069   1071   1076   1078*  1091*  1092   1099   1101   1113*  1116   1120
                       1122    1124   1128*  1129   1132   1135
OUT     017276         3040    3042   3044   3048   3058   3060#
OPDAR   002716         1087    1104   1107#  1123
OPDSEL  006504         1282    1312   1370   1670#
OPPAT   003026         1109    1125#  1130
OPRD    003150         1142    1145#
OPWCK   003200         1148    1151#
OPWRT   003120         1133    1139#
OUT     013466         1036    2467#
OUTBUF  017436         1191*   1198   1200   1230   1277   1316   1336   1442*  1443   1455   1457   1462   1464
                       1610    1883*  2107*  2114*  2118*  2446*  2451   3104#
PARCSR= 172100         2682#   2690
PAREER  000736          766#   2700
PARVEC= 000114         2683#   2684*  2685*
PASEL   007722         1279    1870#  2452
```

```
PASEX   010030          1889#   1925
PASSC   001156           885#   1449#   1563#
PATNU   001136           877#    957#   1131#   1135#   1137#   1414#   1415    1418#   1447#   1566#   1606    1753    1872
                        1993    2003    2016    2023    2442#   2668#
PAT0    000254           626#   1883    1885    2017
PAT1    000256           627#
PAT10   000274           634#
PAT11   000276           635#
PAT12   000300           636#
PAT13   000302           637#
PAT14   000304           638#
PAT15   000306           639#
PAT16   000310           640#
PAT17   000312           641#
PAT2    000260           628#
PAT20   000314           642#
PAT3    000262           629#
PAT4    000264           630#
PAT5    000266           631#
PAT6    000270           632#
PAT7    000272           633#
PCNT    001004           776#   2299#   2300#   2513    2529#   2530#   2538    2641
PDOWN   000721           763#   2465
PFT1    012506           622    2320#
PFT2    013034           623    2392#
PFWAT   012542          2328#   2341
PFWATT  012534          2326#   2362
PHADDR  001120           832#   2114    2230#
PHYCOV  006632          1345    1701#
POWFAL  013312          2327    2397    2442#
PRIORI  001066           816#   1731    2345    2348    2403
PRNT    014224          1245    1293    1331    1341    1394    1478    1505    1549    1678    1792    2042    2046    2084
                        2126    2204    2561    2566#
PS    = 177776           695#    696     932#   1190#   1731#   2051    2345#   2348#   2403#   2707
PSW   = 177776           696#
PTDONE  014740          2617    2621    2634    2638#
PWRDN   013242          2401    2425#   2428
PWRFL   013072          2398    2400#   2416
PWRFL2  013062          2397#   2422
PWRFI   013070          2394    2399#   2434
PWRUP   013252          2425    2428#
QQ    = 000001          1183#   1260#   1434#   1575#
RAND    010244          1938#   2009
RANDM   000674           759#   1480    1507    1551
RANDOM  010044          1456    1463    1880    1894#
RANDI   010060          1896#   1924
RANEL   005054           621    1428#
RANEX   005062          1425    1430#   1570
RDERR   000607           743#   1397    1552    2493
RDLIN = 104422          1169    2994#   3074
RDOCT = 104420          1049    1053    1067    1090    1112    1127    2322    2993#   3055
RDSECT  003526          1219#
READCT  005050          1411    1427#
READER  001206           897#   2495    2521#   2650#   2651
RECOV   000616           746#   1523    2206
RESTOR  010220           965    1931#
```

```
RETT    015172          2639    2647    2654    2670#
RLDR    015174           617    2672#
RORBLK  007652          1849    1853#
RRDSEC  003524          1218#   1253
RREAD   005656          1532#   1557
RRR2    011756          1702    1857    1861    2142    2148    2169    2184    2191#
RSAS    001050           796#    979    1654*   2337*   2355*   2412*   2594
RSBA    001040           792#   1335    1509    1610*   1612*   1723    2599
RSCS1   001032           789#   1207    1209    1222    1224    1631*   1638    1641    1705    1708    2034    2579    3170
                        3174    3197    3201
RSCS2   001034           646#    647*    790#    973*    977*    989*   1603*   2372*   2430*   2587   3166*   3167*   3172
                        3193*   3194*   3199
RSDA    001042           793#   1600*   1784    1790    2604
RSDB    001054           798#   2619
RSDS    001044           794#   1007    1645    2373    2432    2614
RSDT    001060           800#    991     995     997     999    1001    2626
RSER    001046           795#    974*   2338*   2354*   2411*   2583
RSLA    001052           797#   2636
RSMR    001056           799#   2631
RSREG   014234          2568#   2816
RSVEC   001062           801#   1730*
RSWC    001036           791#   1353    1609*   1615*   2609
RSO4DT  001166           889#    990*    993*   1003*   1114    1118    1688    1693    1757    1763    1829    1833    1839
                        1844    1855    1862    2061    2065    2073    2077
RW    = 000006           817#   2242    2243    2244    2245    2246    2247    2248    2249
RWRED   005712          1450    1537    1539    1540#
SANHT   003614          1235#   1239
SAVAST  001072           821#   2107    2108    2143*
SAVCPU  001076           823#   2118    2119    2149*
SAVE    001150           882#   1193*   1254    1989*   2018*   2022*   2036*   2037    2052    2055    2216*   2231    2708
                        2711
SAVMGA  001100           824#   2111    2153*
SAVMGB  001102           825#   2159*
SAVMGC  001104           826#   2122    2160*
SAVWC   001214           900#    953*   1094    1931
SAVWCB  001220           902#    954*   1097    1933
SCOPE = 104400          1186    1215    1218    1229    1251    1263    1306    1310    1365    1374    1400    1406    1437
                        1483    1492    1527    1529    1554    1562    1578    2528    2985#
SEABUF  003426          1199#   1211    1217
SEEC    014622          2590    2622#
SETIND  014150          2548#   2552
SETSWI  001230           607     909#
SHIFT   010076          1900#   1904
SHIFT1  010262          1942#   1946
SIZB1   020742          3200    3217#
SIZB2   020752          3216    3219#
SIZB3   020734          3211    3214#   3218
SIZEAP  001106           827#   2161    3176*   3181*
SIZEBP  001110           828#   2176    3205*   3220*   3232
SIZERR  020572          3177    3203#
SIZP    021164          3230    3241#   3244
SIZPR   021156          3183    3233    3239#
SIZZAP  020210           948    3156#
SIZZBP  020452          3186#
SIZ1    020366          3173    3180#
SIZ2    020544          3196#   3215
```

```
SIZ3     020710            3202    3210#
SLH      004202            1284    1310#
SLH2     004204            1311#   1359    1366    1378
SLH2A    004572            1368    1378#
SR0    = 177572             652#    940*   1446*   2253*
SR1    = 177574             653#
SR2    = 177576             654#
SR3    = 172516             655#
STABUF   000653             755#   1048    1052
STAMEM   001070             820#   1050*   1711    2135    2163    3162*
STATUS   001064             815#   1275*   1452*
STBCOM   001074             822#   1054*   1714    2146    2156    2178    3213*   3227
STMM2    012040            2112    2212#
STTEST   002014             984     987#   1023
SUSWR  = 104444             924    3003#
SWI      020116             909*    922    3110    3128*   3131#
SWITCH   001174             892#   1990*   2007    2011*   2013*
SWR      001026             784#    936     959     962     983    1009    1161    1325    1327    1338    1383    1390    1419
                           1472    1500    1536    1543    1545    1658    1977    2030    2039    2090    2197    2467    2510
                           2566    2643    2646    2713    2766    2777    2807    2817    2820    3041    3115    3119*   3121
SWRDCT   001144             880#    953     958    1057    1101*   1102    1193    1197*   1254*   1266    1805    1931*   1932
                           2171*   2173*   2448*   2449
SWREG    000176             599#   3041    3052    3059*   3119    3121
SW0    = 000001             590#
SW1    = 000002             589#
SW10   = 002000             580#   2531    2805
SW11   = 004000             579#   2777
SW12   = 010000             578#   2531
SW13   = 020000             577#   2807
SW14   = 040000             576#   2766
SW15   = 100000             575#
SW2    = 000004             588#
SW3    = 000010             587#
SW4    = 000020             586#
SW5    = 000040             585#
SW6    = 000100             584#
SW7    = 000200             583#
SW8    = 000400             582#   2765
SW9    = 001000             581#   2820
TDMA     001142             879#   1124*   1672
TIMES    015734            2781    2796#
TIMUP    007016            1738#   2908
TKB      001022             782#   3045
TKS      001020             781#   3043
TPB      001024             783#   2739*   2744*
TPS      001016             780#   2740    2745
TRACK    001132             877#
TRUERR   006442            1644    1646    1657#
TRY      001722             974#    978
TRYNX    002212             988    1002    1008    1020#   1583
TSTNG    000706             761#    985
TST1     003332            1186#
TST2     003716            1263#
TST3     005062            1437#
TST4     006070            1578#
TYPE   = 104402             921     985    1013    1017    1043    1048    1052    1061    1066    1084    1089    1107    1111
```

```
                        1126    1140    1146    1152    1168    1247    1295    1296    1333    1334    1348    1352    1396
                        1397    1480    1481    1507    1508    1511    1518    1523    1526    1551    1552    1680    1795
                        1968    1971    1974    2044    2048    2086    2089    2206    2209    2321    2360    2419    2465
                        2482    2487    2488    2489    2492    2493    2494    2497    2498    2499    2502    2505    2512
                        2515    2571    2574    2578    2582    2586    2593    2598    2603    2608    2613    2618    2625
                        2630    2635    2640    2645    2656    2659    2700    2704    2809    2815    2867    2907    2986*
                        3021    3026    3029    3032    3049    3051    3054    3102    3182    3204    3223    3226    3231
                        3247    3326
TYPEA = 104412          2053    2709    2990*
TYPED = 104446          2989*   3246    3278*
TYPEO = 104404          1351    1354    1513    2573    2576    2580    2584    2588    2595    2600    2605    2610    2615
                        2620    2627    2632    2637    2814    2987*   3053
TYPES = 104406          1016    1031    1525    1797    2056    2088    2208    2484    2491    2496    2501    2504    2514
                        2642    2658    2712    2988*
TYPREC  012012          1304    1363    1409    1491    1561    2204*
UNCMP   001164           888*    981*    987    1020*   1028*   2655
UNITSV  001162           887*    979*    980*    987    1080*   2471*   2472    2507*   2661
UNNUM   001160           647     886*    982*    989    1015    1022*   1029*   1030    1071*   2323*   2372    2430    2556
                        2648    2657    3167    3194
UNRECO  000522           727*   1518    1680
UNTYP   002024           989*   1685
UP      = 000000         818*   2242    2243    2244    2245    2246    2247    2248    2249
UPCHK   012632          2342*   2375
UPDAT = 104424          2995*   3178    3214
UPP     013000          2367    2370*
VADDR   001116           831*   1443*   1878    1989    2099    2108*   2113*   2119*   2447*
VECTRR  007000          1274    1451    1730*   2324    2396
WATT    011770          1287    1319    1382    1469    1497    1535    2197*   2333    2349    2406    2455    2458
WC      = 000010         807*   1214    1228    1998    2286    2606
WCCON   002606          1086    1088*   1093    1100
WCERR   001204           896*   2500    2522*   2651*   2652
WCKERR  000574           740*   1334    1508    2498
WDCTB   001112           829*    954    1105*   1811    1933*   2117    2186*   2188*
WDERR   010712          2019    2028*
WHTHU   006566          1196    1232    1460    1688*
WORK    001222           906*   1346*   1350    1454*   1461*   1509*   1510*   1512    1738*   1743*   1794*   1796    1877*
                        1887*   1919*   1923*   1995*   1996*   2472*   2474
WORK1   001224           907*   1711*   1714*   1715    1723*   1724*   1725*   1726*   1727    1739*   1741*   1774*   1819*
                        2082*   2083*   2087    3161*   3180*   3181    3186*   3213    3219*   3220    3227*   3232*   3243
                        3252*   3256    3258    3262    3267
WORK2   001226           908*   1813*   1815*   1822    1988*   2020*   2060
WRDCMP  010646          2012    2015    2018*   2025    2029
WRDCT   001130           874*    958*   1057*   1102*   1105    1197    1201    1231    1233    1234*   1266*   1461    1613
                        1690*   1692*   1695*   1805*   1811*   1822*   1824*   1825*   1826*   1827*   1828*   1832*   1835*
                        1847    1877    1932*   1988    2058    2100    2117*   2449*   3164*   3191*
WRDINC  010654          2020*   2094
WRDNW   013406          2453*   2464
WRERR   005274          1453    1466*   1486
WRITER  001202           895*   2490    2520*   2650
WRLG1   005172          1453*   1565
WRNEXB  003514          1210    1215*
WRRCK   005434          1494*   1517
WRRCK1  005414          1471    1489*
WRTERR  000564           737*   1296    1481    2488
WRXBL   004146          1290    1302*
WRX1    004160          1303    1305*
```

```
WTNO    011460  1301   1487   2126#
XESH    004250  1320#
XINCSE  007442  1815#  1820
XINCSU  007466  1817   1822#
XSEABU  003440  1202#  1204
XWAIT   012332  2199   2261#
XXX     020106  3111   3128#
YORN    000641   752#  1168
$ENDAD  014110   593   2534#
$END1   014120  2532   2536#
.       = 021614   591#   592#   594#   597#   606#   609#   613#   724#   770#   772#   921    922#  1043
                  1044#  1061   1084   1089   1090#  1111   1126   1140   1146   1147#  1152   1153#  1208
                  1223   1257   1348   1352   1353#  1416   1511   1569   1659   1795   1796#  1968   1971
                  1974   2044   2048   2049#  2086   2087#  2089   2090#  2360   2361#  2419   2420#  2487
                  2488#  2489   2490#  2492   2493#  2494   2495#  2497   2498#  2499   2500#  2502   2505
                  2506#  2512   2515   2516#  2538   2571   2572#  2574   2575#  2578   2582   2583#  2586
                  2593   2594#  2598   2599#  2603   2604#  2608   2609#  2613   2614#  2618   2619#  2625
                  2626#  2630   2631#  2635   2636#  2640   2645   2646#  2656   2704   2705#  2741   2746
                  2796   2809   2810#  2815   2816#  2818   2826   2827   2836   2856   2859   2864   2874#
                  2907   2911   3016   3021   3022#  3026   3036#  3037#  3049   3050#  3051   3054   3055#
                  3102   3107#  3140   3182   3204   3223   3226   3227#  3231   3232#  3247   3331#
.CLRDV  000316   646#  2996
.CNTLU  017214  3001   3050#
.DBLK   021602  3293   3326   3331#
.DKCMD  006146  1600#  2992
.DSIGN  021612  3281#  3287#  3291#  3304   3306#  3308   3332#
.DTBL   021572  3296   3329#
.ERCLR  006124  1590#  2991
.HLT    015736   913   2805#
.HLTCT  016060  2569   2589   2591   2596   2601   2606   2611   2616   2622#  2623   2628   2633   2812#
                2826#
.ILLUP  016422  2883   2910#
.KBDIN  017132  3002   3039#
.KIT    015706  2767   2776   2782   2789#  2824
.LOGC   014156  2551#  3000
.LOGR   014136  2544#  2998
.LOGW   014126  2540#  2997
.LOGWC  014144  2547#  2999
.MAMK   015222   950   2684#
.MSG    017300  3045#  3046#  3047   3056#  3059   3062#
.OPDR   000535   731#  1107
.OVER   015712  2790#
.PARSR  015314  2684   2700#
.POWER  016274   911   2883#  2905
.POWUP  016340  2892   2895#
.PR     016250  2833#  2835#  2837#  2838#  2843   2847   2857#  2858   2861#  2863   2867   2874#  2961#
                2962#  2963   2964#
.PRF    016156  2845   2848   2853#
.PRL    016134  2846#  2862   2966
.PTIT   016114  2834   2839#
.PUVEC  016430  2883#  2884#  2892#  2914#
.QQ.    = 000050  2972#  2985   2986#  2987#  2988#  2989#  2990#  2991#  2992#  2993#  2994#  2995#  2996#
                  2997#  2998#  2999#  3000#  3001#  3002#  3003#  3004#  3278   3279#
.RDLIN  016730  2994   3011#
.RDOCT  017304  2993   3069#
.SATAB  016632  2937   2968#
```

```
.SAVR6  016426          2891*   2895    2913*
.SCOPE  015562          2765*   2985
.SUSWR  020000          3003    3110*
.SVLAD  015666          2773    2778    2784*   2792
.SWOPT= 121000           569*    575     576     577     578     579     580     581     582    2531    2765    2805    2807
                        2820
.TBIT   014124          2538*
.TRAP   016642           915    2979*
.TRP    016660          2982    2983*
.TYPE   015424          2728*   2986
.TYPEA  016434          2922*   2990
.TYPEB  016062          2833*
.TYPED  021364          2989    3278    3281*
.TYPEO  016072          2835*   2987
.TYPES  016102          2837*   2988
.UPDAT  021216          2995    3252*
```

```
END          1#    569#
LDPDR     2211#   2242    2243    2244    2245    2246    2247    2248    2249
NEWTST       1#    569#   1183    1260    1434    1575
ODT11        1#
ODT11X       1#
POP          1#    722#   2772    2775    2868    2899    2965    3096    3319
PRINT        1#    722#    921    1043    1061    1084    1069    1111    1126    1140    1146    1152    1348    1352    1511
          1795    1968    1971    1974    2044    2048    2086    2089    2360    2419    2487    2489    2492    2494    2497
          2499    2502    2505    2512    2515    2571    2574    2578    2582    2586    2593    2598    2603    2608    2613
          2618    2625    2630    2635    2640    2645    2656    2704    2809    2815    2907    3021    3025    3049    3051
          3054    3101    3182    3204    3223    3226    3231    3247
PUSH         1#    722#   2769    2839    2885    2922    2931    3071    3282
SCOPE.       1#   1183#   1260#   1434#   1575#
SET       2972#   2985    2986    2987    2988    2989    2990    2991    2992    2993    2994    2995    2996    2997    2998
          2999    3000    3001    3002    3003    3278
TYPADR       1#
TYPBIT       1#    722#
TYPCHR       1#    722#
TYPDEC       1#    722#   3245
TYPLIN       1#    722#
TYPOCS       1#    722#   1014    1030    1796    2087    2483    2490    2495    2500    2503    2513    2641    2657
TYPOCT       1#    722#   1349    1353    1512    2572    2575    2579    2583    2587    2594    2599    2604    2609    2614
          2619    2626    2631    2636    2813    3052
TYPTXT       1#    722#    921    1042    1061    1083    1088    1111    1125    1139    1145    1151    1347    1352    1511
          1795    1967    1971    1973    2044    2048    2086    2089    2360    2419    2487    2489    2492    2494    2497
          2499    2502    2505    2512    2515    2571    2574    2577    2581    2585    2593    2598    2603    2608    2613
          2618    2625    2630    2635    2640    2645    2656    2704    3049    3050    3054    3182    3203    3223    3225
          3231    3247
$CATCH       1#    569#    591
$CMTAG       1#    569#    772
$DONE        1#    569#   2526
$EQUAT       1#    569#    693
$HLT         1#    569#   2797
$KBDIN       1#    569#   3038
$KMMR        1#    569#    650
$OCTAL       1#    569#   2827
$POWER       1#    569#   2875
$RAND2       1#
$RAND4       1#
$RDDEC       1#
$RDLIN       1#    569#   3004
$RDOCT       1#    569#   3064
$SCOPE       1#    569#   2758
$SET      2972#   2985    2986    2987    2988    2989    2990    2991    2992    2993    2994    2995    2996    2997    2998
          2999    3000    3001    3002    3003    3278
$SETUP       1#    569#    910
$SMMR        1#
$SUSWR       1#    569#   3109
$SWDOC       1#    569#    572
$TRAP        1#    569#   2972
$TYPE        1#    569#   2718
$TYPEA       1#    569#   2915
$TYPED       1#    569#   3279
$UMMR        1#
```

. ABS. 021614 000

ERRORS DETECTED: 0

CZRSCG.BIN,CZRSCG.LST/CRF/SOL/NL:TOC=CZRSCG.SML,CZRSCG.P11
RUN-TIME: 8 12 1 SECONDS
RUN-TIME RATIO: 74/21=3.4
CORE USED: 22K (43 PAGES

EOF1C2RSCGSE3 00010000 780330 PDP10 411