DataGeneral

TECHNICAL STATEMENT

TEXT LISTING

068-000271-05

PROGRAM

ECLIPSE MULTI-PROGRAMMING
RELIABILITY (LONG)

TEXT TAPE

097-000271-05

ABSTRACT

THE ECLIPSE MULTI-PROGRAMMING RELIABILITY TEST (LONG) CONSISTS OF A SERIES OF TESTS AND A SUPERVISOR PROGRAM (THE DIAGNOSTIC LINKER). THIS VERSION INCLUDES THOSE TESTS THAT APPLY TO THE CPU, MEMORY, FLOATING POINT AND THE MMPU PLUS PRIMARY DEVICE CODE TESTS FOR THE NOVA DISK, MOVING HEAD DISK, MAGNETIC TAPE, CASSETTE TAPE AND LINE PRINTER.

COPYRIGHT © DATA GENERAL CORPORATION, 1975, 1976, 1977, 1978
ALL RIGHTS RESERVED. PRINTED IN U.S.A.

```
01
02
03
04
05
06
07          ;*************************************************************
08          ;
09          ; NAME: EMORTL.TX                          PART NUMBER: 097-000271
10          ;
11          ; DESCRIPTION: ECLIPSE MULTI-PROGRAMMING RELIABILITY
12          ;               LONG VERSION
13          ;
14          ; REVISION HISTORY:
15          ;
16          ;       REV.            DATE
17          ;
18          ;       04              09/01/77
19          ;       05              10/06/78
20          ;
21          ; COPYRIGHT © DATA GENERAL CORPORATION, 1975, 1976, 1977, 1978
22          ; ALL RIGHTS RESERVED.
23          ;*************************************************************
```



```
01                  PATCH=0
02                  ;FILE FOR EMORTV1 CPU AND PRIMARY DEVICES
03          000000 SHORT=0
04          000000 ERCOR=0
05          000001 ERCCF=1
06          000000 CBRDS=0
07          000000 ADRTS=0
08          000000 EISTS=0
09          000000 ARITH=0
10          000000 EATS=0
11          000000 FPUTS=0
12          000000 LEFTS=0
13          000000 COMER=0
14          000000 SCMTS=0
15          000000 DCUTS=0
16          000000 WCSTS=0
17          000000 PGDSK=0
18          000000 NVDSK=0
19          000000 MVDSK=0
20          000000 PZDSK=0
21          000000 MTTES=0
22          000000 CATES=0
23          000000 LPTTS=0
24          000001 P.DSK=1
25          000001 N.DSK=1
26          000001 SZDSK=1
27          000001 MXDSK=1
28          000001 CXTES=1
29          000001 MXTES=1
30          000001 IOTST=1
31          000000 PITTS=0
32          000000 .DUSR .MAPD=0
33          000001 ZLOAD=1
```

```
                ;       IF THE MOVING HEAD DISKS ARE TO BE
                ;       EXERCISED THEY MUST HAVE A PACK INSTALLED
                ;       AND BE IN THE READY STATE
                ;       IF MAGNETIC TAPES ARE TO BE EXERCISED
                ;       THEY MUST BE ON LINE WRITE ENABLED
                ;       IF CASSETTES ARE TO BE EXERCISED
                ;       THEY MUST BE ON LINE WRITE ENABLED
                ;       IF THE LINE PRINTER IS TO BE EXERCISED
                ;       IT MUST BE ON LINE AND IN THE READY STATE
                ;       IF THE DCU 50 TEST IS TO BE RUN AT A DEVICE CODE
                ;       OTHER TAN 64,ONE LOCATION HAS TO BE CHANGED SUCH
                ;       IT CONTAINS THE DCU 50 DEVICE CODE.SEE DCU-50 TEST
                ;       LOCATION DCUDV:
EEU     000000  PATCH=DCUDV
                ;
                ;3.      OPTIONAL STARTING ADDRESSES
                ;3.1     200   AUTO-SIZE AND GO START
                ;3.2     202   MANUAL SELECT/DELETE TESTS START
                ;3.3     204   RUN UP TO FIRST 32K UNMAPPED START
                ;3.4     206   RESTART LAST TEST SELECTIONS
                ;3.5     210   START IN THE ODT
                ;
                ;       NOTE:   THIS PROGRAM CONTAINS A OCTAL DEBUGGER
                ;       TO ASSIST IN GATHERING INFORMATION AT
                ;       TIME OF ERROR AND PATCHING THE PROGRAM.
                ;
                ;         IF STARTED FROM DTOS AND THIS PROGRAM
                ;       VERSION IS EMORT L OR EMORT P THEN IT
                ;       WILL AUTOMATICALLY START AT ADDRESS
                ;       210 TO ALLOW SELECTION OF TESTS/PATCHING
                ;       AND OR OPERATOR SETUP. TYPE P TO SELECT
                ;       A 202 START, OR TYPE 204R OR 200R FOR
                ;       OTHER STARTS.
                ;
```



```
                ;3.6    KEY ENTERED SWITCH OPTIONS(SWREG)
                ;
                ;               THE UTILIZATION OF KEYENTERED OPTIONS
                ;               HAS BEEN IMPLEMENTED SUCH THAT THE
                ;               CONSOLE "MONITOR" FUNCTION CAN
                ;               BE FULLY UTILIZED WITHOUT INTERFERING
                ;               WITH PROGRAM FLOW.
                ;
                ;  KEY 0        LOCKS THE SWPACKAGE INTO INPUT MODE
                ;               ALLOWING SETUP OF THE CONTENTS OF "SWREG".
                ;               TYPE A CARRAGE RETURN TO EXIT.
                ;
                ;       TYPING KEY'S 1 - 9,A - F SETS/RESETS SWREG BITS
                ;       1 - 9,10 - 15 RESPECTIVELY. EACH KEY ENTRY
                ;       COMPLEMENTS THE PREVIOUS STATE OF THE SWREG BIT.
                ;
                ; KEY   SWREG BIT       FUNCTION
                ;
                ; 1     1 = 1           DELETE TTO TYPEOUTS
                ;
                ; 2     2 = 1           DELETE MEM ALLOCATION TABLE
                ;                       FROM TYPEOUTS
                ;
                ; 3     3 = 1           INCREASES THE CHANCE OF SELECTION
                ;                       OF THE TTY AND LPT TEST.
                ;
                ; 4     4               TYPING A 4 WILL CAUSE THE ELAPSED RUN
                ;                       TIME AND ACCUMULATED ERRORS
                ;                       TO BE TYPED ON THE TTY.
                ;                       (NOTE: A RTC MUST EXIST)
                ;
                ; 5     5 = 1           DIRECT ERROR AND RUNTIME TYPEOUTS
                ;                       ALSO TO THE LINE PRINTER .
                ;
                ; 6     6 = 1           THE ERROR ROUTINE WILL PAUSE AFTER
                ;                       EACH PHASE OF AN ERROR TYPEOUT.
                ;                       TYPE A CR KEY ON DEVICE TTI TO PROCEED.
                ;
                ; 7     7               TYPING A 7 WILL CAUSE
                ;                       INDIVIDUAL RUN STATISTICS OF
                ;                       EACH TEST TO BE LISTED.
                ;
                ; F     15 = 1          DOESN'T RELEASE OR ALLOW REASIGNMENT
                ;                       OF SCRATCH AREAS AFTER ERROR.
                ;
                ;       NOTE (C) = CONTROL KEY
                ;
                ;
                ;       KEY M     PRINTS THE CURRENT CONTENNTS OF SWREG.
                ;       KEY (C)D  RESTARTS THE PROGRAM AT LOC 202
                ;                 AFTER RESETTING SWREG TO ALL 0'S
                ;       KEY (C)R  RESTARTS THE PROGRAM AT LOC 202
                ;                 WITHOUT DISTURBING SWREG.
                ;
                ; EXAMPLE:
                ;       TO DELETE ERROR TYPEOUTS AND LOOP ON FAILING
                ;       SCRATCH AREAS TYPE 0,1,F, AND CR.
```

```
!0005  .MAIN
                ;4.        OPERATING PROCEDURES
                ;4.1       LOAD THE PROGRAM VIA THE BINARY LOADER
                ;4.2       SET SWITCHES TO:
                ;        200 FOR AUTO SIZE AND GO
                ;        202 FOR MANUAL SELECT/DELETE
                ;        204 TO IGNORE MMPU
                ;        206 TO RESTART LAST SELECTED TESTS
                ;NOTE:   THE TEST MUST HAVE BEEN STARTED
                ;        AND RUN FOR A FEW SECONDS
                ;        BEFORE ADRS 206 MAY BE UTILIZED
                ;4.3       PRESS START
                ;4.4       PROCESSOR WILL TYPE:
                ;        ECLIPSE MULTI-PROG. REL. TEST XXX VERSIION
                ;        TOTAL #1K'S=XXX(DECIMAL) MMPU,MMPU1 OR NO MAP
                ;        PROGRAM RUN LIST
                ;        PROG# DESCRIPTION
                ;4.5       IF START WAS 200 (OR 206) THE LIST OF
                ;        PROGRAMS TO BE RUN CONCURRENTLY WILL
                ;        THEN BE LISTED AND THE TEST SYSTEM
                ;        WILL AUTO START
                ;4.6       IF START WAS 202 LINKER WILL
                ;        PAUSE AT THE END OF EACH TEST
                ;        DESCRIPTION AND WAIT FOR KEYBOARD
                ;        INPUT.  TYPING IN A SPACE WILL
                ;        ENABLE THAT TEST TO BE RUN.
                ;        TYPING IN ANY OTHER CHARACTER WILL
                ;        DELETE THAT TEST FROM BEING RUN
                ;4.7       IF THE STARTING ADDRESS WAS 204 THE LINKER
                ;        WILL SIZE MEMORY WITHOUT UTILIZING
                ;        OR EVEN LOOKING FOR THE MAP OPTION
                ;        AND THEN PROCEED AS IN STARTING AT ADRS. 202
                ;        WITH THE MAP NONEXISTENT.
                ;4.8     IF STARTED IN ODT (ADDRESS 210) AND "P"
                ;        IS TYPED THE TEST WILL BE STARTED AS IF
                ;        IT WAS STARTED AT 202.
                ;4.9     OPTION SET UP
                ;        IF A AUTOSTART ADDRESS WASN'T USED
                ;        THEN THE PROGRAM WILL PAUSE AFTER PRINTING
                ;        "ENTER OPTIONS,CR TO CONTINUE" AND ALLOW
                ;        KEY ENTRY OF SWITCH REG. OPTIONS. TYPE
                ;        A "CR" KEY TO START TESTS.
                ;
```

```
!0006  .MAIN
                ;4.10   DUAL PORT DISK START UP PROCEDURE
                ;
                ; 4.10.1        PROGRAM LOAD EMORT(LONG OR PERIPHERAL)
                ;               INTO EACH CPU.
                ;
                ; 4.10.2        INSTALL APPROPRIATE ODD/EVEN CYLINDER
                ;               PATCHES INTO THE PROGRAM FOR EACH CPU.
                ;               (SEE PARAG. 4.10.6)
                ;
                ; 4.10.3        START ONE CPU AT 202 ;WHEN QUERRED
                ;               ABOUT TEST DELETIONS, START OTHER
                ;               CPU AT 202.
                ;
                ; 4.10.4        WHEN QUERRED ABOUT TEST DELETIONS
                ;               ON SECOND CPU ,ENTER ALL TEST
                ;               SELECTIONS ON BOTH CPU'S.
                ;
                ; 4.10.5        ENTER ANY SWREG OPTIONS. CARRIAGE
                ;               RETURN STARTS THE PROGRAM.
                ;
                ; 4.10.6        PATCH LIST
                ;
                ; 4.10.6.1      PRIMARY DEVICE CODE DISKS
                ;
                ;
                ;LOCATION LABEL  TEST            ONE SYS OTHER SYS
EEU       000006 PATCH=  ND.3L+6 ;NOVA DISK      101232  101233
EEU       000006 PATCH=  PD.3L+6 ;6063/64 DISK   101232  101233
EEU       000006 PATCH=  MH.3L+6 ;MOVING HEAD    101232  101233
EEU       000006 PATCH=  ZB.3L+6 ;6060/61 DISK   101232  101233
                ;
                ; 4.10.6.2      SECONDARY DEVICE CODE DISKS
                ;
                ;LOCATION LABEL TEST             ONE SYS OTHER SYS
                PATCH=  NX.3L+6 ;NOVA DISK      101232  101233
                PATCH=  PX.3L+6 ;6063/64 DISK   101232  101233
                PATCH=  MV.3L+6 ;MOVING HEAD    101232  101233
                PATCH=  ZS.3L+6 ;6060/61 DISK   101232  101233
```

```
                ;
                ;4.11   DATA LATE RETRIES FOR DISK TESTS
                ;
                ; DUE TO IMPLEMENTATION OF DISK CONFIGURATIONS WHICH WILL
                ;RECEIVE DATA LATE STATUS ERRORS IN THEIR NORMAL COURSE
                ;OF OPERATION, IF THEY ARE SECONDARY PRIORITY ON THE DATA
                ;CHANNEL BUSS TO A DISK WHICH IS A FAST ACCESS TYPE, MULTI-
                ;PROGRAMMING DISK TESTS NOW ARE CODED TO RETRY UP TO 16 TIMES
                ;UPON DETECTION OF A DATA LATE ERROR. IF AFTER 16 RETRIES THE
                ;ERROR STILL CONTINUES, THE ERROR WILL BE REPORTED. ALL
                ;DATA LATE ERRORS WILL BE REPORTED AT SYSTEM STATUS REPORT
                ;TIME, I.E. WHEN A "7" IS TYPED ON THE CONSOLE.
                ;
                ;4.11.1 PATCHES TO DISABLE DATA LATE RETRIES
                ;
                ; THE FOLLOWING PATCHES SHOULD BE INSTALLED IN THE DISK
                ;TESTS WHICH LEGALLY SHOULDN'T RECEIVE ANY DATA LATE ERRORS.
                ; THIS WILL DISABLE THE RETRY FEATURE FOR THAT TEST.
                ;
                ;4.11.1.1 PRIMARY DEVICE CODE DISK TESTS
                ;
                ;LOCATION LABEL TEST            WAS     CHANGE TO
EEU     000000 PATCH=NDRTY      ;NOVA DISK      1000    401
EEU     177777 PATCH=PD.SS=1    ;6063/64 DISK   403     401
 UF     177777;PATCH=MHRWI+8.   ;MOVING HEAD    402     401
 UF     177777;PATCH=ZBRTY      ;6060/61 DISK   402     401
                ;
                ;4.11.1.2 SECONDARY DEVICE CODE DISK TESTS
                ;
                ;LOCATION LABEL TEST            WAS     CHANGE TO
                ;
                PATCH=NXRTY     ;NOVA DISK      1000    401
                PATCH=PX.SS=1   ;6063/64 DISK   403     401
                PATCH=MVRWI+8.  ;MOVING HEAD    402     401
                PATCH=ZBRTY     ;6060/61 DISK   402     401
```



```
                ;5.     ERROR DESCRIPTION
                ;       MOST ERRORS DETECTED BY EITHER
                ;       THE INDIVIDUAL TEST PROGRAMS OR
                ;       BY THE DIAGNOSTIC LINKER WILL
                ;       RESULT IN AN EXTENSIVE ERROR
                ;       TYPEOUT.  SOME SMALL NUMBER OF
                ;       HIGHLY IMPROBABLE ERRORS MAY RESULT
                ;       IN A PROGRAM HALT IF THEY ARE
                ;       OF A NATURE THAT THE LINKER CANNOT
                ;       RECOVER FROM AND LOGICALLY PROCEED,
                ;       (I.E. INTERRUPT STACK OVERFLOWS)
                ;NOTE:  AFTER 50 ERRORS HAVE BEEN DETECTED, THE
                ;       TEST WILL INDICATE SUCH VIA TYPEOUT, THEN
                ;       WAIT FOR A CR KEY FROM DEVICE TTI IF TYPEOUTS
                ;       AREN'T DELETED(SW1=1)

                ;5.1    ERROR FORMAT
                ;       ERROR TYPEOUTS INCLUDE
                ;5.1.1  PROGRAM # AND NAME AT TIME OF ERROR
                ;5.1.2  THE CURRENT CONTENT OF THE USER STACK
                ;       IN UP TO 2 SETS OF 5 VARIABLES.
                ;       EACH SET OF VARIABLES (1 STATE BLOCK) INCLUDE
                ;       AC0 TO AC3 AND THE PC WITH CARRY.
                ;NOTE: THE 2 STATE BLOCKS TYPED ARE THE FIRST
                ;       AND LAST 5 ENTRIES ON THE USER STACK.
                ;       IF THERE ARE SAY SEVEN ENTRIES ON THE STACK
                ;       THESE STATE BLOCKS WILL OVERLAP AND FOR THIS
                ;       REASON THE START ADRS OF EACH STATE BLOCK
                ;       IS INCLUDED IN THE TYPEOUT
                ;5.1.3  LOGICAL SCRATCH AND DATA CHANNEL LIMITS
                ;5.1.4  MEMORY ALLOCATION TABLE
                ;       PHYSICAL 1K PAGE# +LOGICAL ADDRESS
                ;       +REL LOG. ADDR(IF RELOCATED)
                ;5.1.5  CONTINUATION INFORMATION IN GROUPS
                ;       OF 3 MEMORY LOCATIONS PERTINENT TO
                ;       THE INDIVIDUAL TEST THAT FAILED
                ;
                ;5.1.6  THE CPU TESTS THAT RELOCATE/REMAP WILL INCLUDE
                ;IN THEIR ERROR TYPEOUTS:
                ;ST.LA  START/ERROR (RES.)
                ;XXXXXX YYYYYY  ZZZZZZ
                ;
                ;ST.LA  THE LOGICAL START OF THE RELOCATED TEST LOOP
                ;XXXXXX (I.E. THE LAST LCALL SETUL)
                ;
                ;START  THIS NUMBER INDICATES WHERE THE RESIDENT COPY
                ;YYYYYY OF THE TEST LOOP MAY BE FOUND IN THE LISTING
                ;
                ;ERROR  THIS NUMBER INDICATES WHERE IN THE RESIDENT
                ;ZZZZZZ COPY OF THE LISTING THE ERROR CALL MAY BE FOUND
                ;       (FOR SOME VALIDITY TRAP ERRORS THIS NUMBER
                ;       MAY NOT APPEAR TO BE VALID.)
```

```
!0009  .MAIN
01              ;5.1.7  FLOATING POINT AC'S
02              ;
03              ;       ALL CPU TESTS WHICH MAKE USE OF THE FLOATING
04              ; POINT WILL IN THEIR ERROR OUTPUT INCLUDE THE CURENT
05              ; STATE OF THE FPSR AND FPAC0 THRU FPAC3 IN THE FOLLOWING
06              ; FORMAT:
07              ;
08              ; FPSR = SSSSSS SSSSSS
09              ;          FPAC0-3
10              ; 0    15 16   31 32   47 48   63
11              ; XXXXXX XXXXXX XXXXXX XXXXXX
12              ; YYYYYY YYYYYY YYYYYY YYYYYY
13              ; ZZZZZZ ZZZZZZ ZZZZZZ ZZZZZZ
14              ; QQQQQQ QQQQQQ QQQQQQ QQQQQQ
15              ;
16              ; WHERE XXXXXX = FPAC0
17              ;       YYYYYY = FPAC1
18              ;       ZZZZZZ = FPAC2
19              ;       QQQQQQ = FPAC3
20              ;
21              ;5.1.8  RELOCATED CODE ERROR
22              ;
23              ; UPON DETECTION OF AN ERROR BY A RELOCATED TEST
24              ; THE RELOCATED CODE IS COMPARED TO THE ORIGINAL
25              ; COPY. IF A DIFFERENCE IS FOUND THE FOLLOWING
26              ; INFORMATION IS TYPED:
27              ;
28              ; RELOCATED CODE ERROR
29              ; EXPECTED ACTUAL ADDR-E ADDR-A
30              ; XXXXX YYYYY QQQQQ ZZZZZ
31              ;
32              ; WHERE,
33              ;  XXXXX  IS THE ORIGINAL WORD
34              ;  YYYYY  IS THE RELOCATED WORD
35              ;  QQQQQ  IS THE ADDRESS OF ORIGINAL
36              ;  ZZZZZ  IS THE ADDRESS OF RELOCATED WORD
37              ;
38              ; WHEN THIS OCCURS THE ERROR WAS
39              ; PROBABLY CAUSED BY THE MODIFICATION OF THE
40              ; RELOCATED CODE.
```

```
!0010  .MAIN
01              ;
02              ;5.2    ERROR ANALYSIS
03              ;       DUE TO THE INTERACTIVE NATURE OF
04              ;       THE TESTS INVOLVED, A SERIES OF
05              ;       ERROR TYPEOUTS WILL PROBABLY BE
06              ;       REQUIRED FOR ANALYSIS BEFORE A
07              ;       PROBLEM WILL BE ISOLATED.
08              ;       A RESTART AT 202 AND DELETION OF ALL
09              ;       BUT THE TEST THAT ORIGINALLY
10              ;       FAILED MAY HELP TO ISOLATE
11              ;       INTERACTIVE PROBLEMS AS FOLLOWS:
12              ;
13              ;       IF THE TEST RUNS BY ITSELF THE PROBLEM
14              ;       IS INTERACTIVE-RE-ENABLE ONE OTHER TEST AT
15              ;       A TIME TO DETERMINE WHICH ONE IS THE PROBLEM.
16              ;       IF THE TEST DOES NOT RUN BY ITSELF
17              ;       RESORT TO SIMILAR BUT LOWER LEVEL TESTS
18              ;       FOR ISOLATION
19              ;
20              ;5.3    PERTINENT MEMORY LOC'S TYPED
21              ;
22              ;5.3.1  CHECKERBOARD RAN
23              ;
24              ;               THE AC'S AT ERROR WILL INDICATE:
25              ;       GOOD DATA- BAD DATA-LOGICAL ADDRESS
26              ;
27              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
28              ;       CB.TK   TEST COUNTER
29              ;       0       GENERATE CHECKERBOARD
30              ;       1       DISTURB PASS
31              ;       2       CHECK PATTERN
32              ;       3       CHECKSUM THE # OF -1'S IN PATTERN
33              ;       CB.LC   STARTING LOGICAL ADDRESS OF "BEGIN"
34              ;               RELOCATED TO SCRATCH
35              ;       CB.SE   AC3 AT ERROR CALL
```

```
;5.3.2  EIS/MRI TEST
;
;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
;       TIME OF ERROR DETECTION
;
;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
;       EI.TK   SEE DISCUSSION OF TEST FOR THE
;               SEQUENCE BEING EXECUTED
;       EI.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
;               REMAPPED FOR EXECUTION
;       EI.LA   LOGICAL START OF TEST AFTER REMAP
;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
;
;5.3.3  ARITHMETIC TEST
;
;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
;       TIME OF ERROR DETECTION
;
;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
;       AT.LC   STARTING ADDRESS OF ARITH IN SCRATCH
;       AT.LO   LOW LIMIT OF SCRATCH AREA AFTER IT IS
;               REMAPPED FOR EXECUTION
;       AT.LA   AT.LC IN RELATION TO AT.LO
;               (LOGICAL START OF ARITH AFTER REMAPPING)
;       THE LAST THREE RANDOM NUMBERS GENERATED
;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
;
;5.3.4  ADDRESS TEST
;
;       THE AC'S 0 1 AND 2 WILL
;        BE TYPED AS THEY
;       WERE AT THE TIME OF ERROR
;       AC0 WILL CONTAIN THE ADDRESS THAT COMPARE WILL
;       STOP AT (AC0<AC2=COMPARE IS TOP TO BOTTOM)
;       AC1 WILL CONTAIN THE DATA FOUND TO BE INCORRECT
;       AC2 WILL CONTAIN THE ADDRESS OR COMPLIMENT
;               OF THE ADDRESS THAT FAILED
;       AC3 WILL CONTAIN THE CURRENT RANDOM OFFSET
;       IN ADDITION THE FOLLOWING LOC.'S WILL BE TYPED:
;       A.TSK   TEST COUNTER
;       0       ADRS. TO ADRS LOADED LOW TO HIGH
;       2       COM. ADRS TO ADRS LOW TO HIGH
;       4       ADRS. TO ADRS LOADED HIGH TO LOW
;       6       COM. ADRS TO ADRS LOADED HIGH TO LOW
;       10      ADRS TO ADRS LOADED LOW TO HIGH
;       1,5, AND 11 COMPARE EACH ADRS. TO EQUAL ITSELF
;       3 AND 7 COMPARE EACH ADRS. TO EQUAL ITS COM.
;       AD.ST   STARTING ADDRESS OF TEST IN CORE
;       AD.S3   ADRS. (AC3) OF ERROR CALL
```



```
;5.3.5  EXTENDED ADDRESSING TEST TEST
;
;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
;       TIME OF ERROR DETECTION
;
;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
;       LA.LC   START ADRS. OF TEST
;               FOR RELOCATION
;       LA.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
;               REMAPPED FOR EXECUTION
;       LA.LA   LOGICAL START OF TEST AFTER REMAP
;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
;5.3.6  FLT PT TEST
;
;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
;       TIME OF ERROR DETECTION
;
;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
;       FP.LC   START ADRS. OF FLT. PT. TEST
;               FOR RELOCATION
;       FP.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
;               REMAPPED FOR EXECUTION
;       FP.LA   LOGICAL START OF TEST AFTER REMAP
;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
;
;5.3.7  LEF/ERROR TEST
;
;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
;       TIME OF ERROR DETECTION
;
;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
;       LF.LC   START ADRS. OF TEST
;               FOR RELOCATION
;       LF.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
;               REMAPPED FOR EXECUTION
;       LF.LA   LOGICAL START OF TEST AFTER REMAP
;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
```

```
;5.3.8  SC MEMORY TEST
;
; THIS IS AN ISZ/DSZ TEST FOR SC-MEMORIES.
;
; THE AC'S AT ERROR WILL INDICATE:
; ACTUAL-EXPECTED-LOGICAL ADDRESS
;
; IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
;       MM.TK   ERROR NUMBER:
;       0       PATTERN STORING ERROR(SHD BE -1)
;       1       LOCATION NOT -1 BEFORE DOING ISZ
;       2       ISZ DIDN'T SKIP
;       3       LOCATION NOT EQUAL TO 0 AFTER ISZ
;       4       DSZ SKIP ERROR
;       5       DSZ TEST-LOCATION NOT -1 AFTER DSZ
;       6       SAME AS 1, EXCEPT TESTING IN REV DIRECTION
;       7       SAME AS 2, EXCEPT   "     "  "    "
;       10      SAME AS 3, EXCEPT   "     "     "     "
;       MM.SE   INSTRUCTION ADDRESS FOLLOWING ERROR CALL
;       LOCATION  ADDRESS OF FAILING LOCATION(LOGICAL)
;
;5.3.9  DCU TEST
;
;       THIS IS AN ARITHMETIC TEST PERFORMED BY THE
;       DCU-50 USING THE DATA CHANNEL.
;
;       THE AC'S AT THE TIME OF ERROR DETECTION WILL
;       BE TYPED .
;       IN ADDITION THE FOLLOWING DATA IS TYPED:
;        RANDOM DATA AC0,AC1,AC2
;        DCLPK  LOGICAL START OF LOOP
;        DCLER  LOGICAL ERROR ADDR
;        DC.LA  LOGICAL START OF TEST
;        DC.LP  LISTING START OF TEST
;        ERROR  LISTING ADDR OF ERROR
```



```
;5.3.10 WCS TEST
;
;       THE AC'S WILL BE TYPED AS THEY WERE AT THE TIME
;       OF ERROR DETECTION .
;
;
;       IN ADDITION THE FOLLOWING LOCTIONS ARE TYPED:
;
;       WS.LC   STARTING ADDRESS OF WCS TEST IN SCRATCH
;       WS.LO   LOW LIMIT OF SCRATCH AREA AFTER IT IS
;               REMAPPED FOR EXECUTION.
;       WS.LA   WS.LC IN RELATION TO WS.LO
;               (LOGICAL START OF WCS TEST AFTER REMAPPING)
;       THE LAST THREE NUMBERS GENERATED
;       (SEE DISCUSSION OF ST.LA,ETC. AT PARAGRAPH 5.1.6)
;
;5.3.11 COMMERCIAL INSTRUCTION TEST
;
;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
;       TIME OF ERROR DETECTION
;
;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
;       CM.LC   START ADRS OF TEST(RELOCATED)
;       CM.LO   LOW LIMIT OF SCRATCH AREA AFTER IT
;               WAS REMAPPED FOR EXECUTION
;       CM.LA   LOGICAL START OF TEST AFTER REMAP
;       (SEE DISCUSSION OF ST.LA,ETC. AT PARA. 5.1.6)
```

```
!0015  .MAIN
                ;5.3.12 NOVA DISK
                ;
                ;       AC0     GOOD DATA
                ;       AC1     BAD DATA
                ;       AC2     ADRS. OF GOOD DATA (BAD IS AT AC2+4)
                ;       ND.SA   MAY= ANY OF THE FOLLOWING
                ;               A. 0,1,2 OR 3 THE ERROR OCCURRED IN ONE
                ;               OF THE FIRST 4 WORDS IN THE BUFFER
                ;               0=1ST 1=2ND 2=3RD 3=4TH
                ;               B. =AC2 ERROR WAS DISK STATUS
                ;               C. A -# DATA ERR IS AT AC2+4
                ;       NDDST   DATA START IN CORE
                ;       NDCST   DATA START FOR DCH MAP (IF MMPU1 EXISTS)
                ;       NDSTA   LAST DISK STATUS
                ;       NDADR   START SECTOR # OF THESE EXERCISED
                ;       ND.CO   1=WRITE, 0=READ
                ;
                ;
                ;       IF OPERATION WAS A READ THEN FOLLOWING
                ;       ADDITIONAL INFORMATION IS TYPED:
                ;
                ;       WRITE NVCST     CHANNEL START ADDR. FOR WRITE
                ;       MD1             FIRST MEM 1K USED IN WRITING
                ;       MD2             NEXT  "   "  "     "  "
                ;       MD3               "   "  "  "     "  "
                ;       MD4               "   "  "  "     "  "
                ;
                ;*******************************************
                ;       NOVA DISK STATUS WORD
                ;*******************************************
                ;  BITS         MEANING
                ;
                ;  0-6          NOT USED
                ;  7            SHIFT REG BIT 0
                ;  8            FIRST BUFFER FULL
                ;  9            SECOND BUFFER FULL
                ; 10            WRITE DATA
                ; 11            WRITE LOCK
                ; 12            DATA LATE
                ; 13            NO SUCH TRACK
                ; 14            ECC ERROR
                ; 15            ERROR
                ;*********************************************
```

```
!0016  .MAIN
                ;5.3.13 6063/6064 DISK
                ;
                ;       AC0     GOOD DATA
                ;       AC1     BAD DATA
                ;       AC2     ADDRESS OF GOOD DATA(BAD IS AT AC2+4)
                ;       PD.SA   MAY=ANY OF THE FOLLOWING
                ;               A) THE ADDRESS OF A RANDOM DATA CONTROL
                ;               WORD(PDDW1 TO PDDW4).THE ERROR WAS IN
                ;               ONE OF THE FIRST 4 WORDS IN THE BUFFER.
                ;               B) A NEGATIVE # ,DATA ERROR IS AT AC2+4
                ;               C) = AC2 ,ERROR WAS A DISK STATUS ERR
                ;               IN WHICH CASE:
                ;               AC2= DISK DIC STATUS
                ;               AC0= 0
                ;
                ;       PDDST   DATA START IN CORE
                ;       PD.CA   LOGICAL ADDR OF CHANNEL IN 1K'S OCTAL
                ;       PDSTA   LAST DISK STATUS
                ;       PDADR   DRIVE+TRACK+SECTOR(FIRST WORD OF QUEUE)
                ;       PD.CO   0=READ,1=WRITE,2=DATA VERIFY
                ;
                ; ALSO IF THE ERROR OCCURS IN A READ OPERATION
                ; THE FOLLOWING DATA IS PRINTED:
                ;
                ;   WRITE PDCST = XXXXX
                ;       MD1     MD2     MD3     MD4
                ;       GGGG    HHHH    JJJJ    KKKK
                ;
                ; WHERE,
                ;       XXXXX = STARTING CHANNEL ADDRESS OF WRITE OPERATION
                ;        GGGG = FIRST PHYS 1K USED IN DISK WRITE
                ;        HHHH = 2ND    "    "  "   "   "    "
                ;        JJJJ = 3RD    "    "  "   "   "    "
                ;        KKKK = 4TH    "    "  "   "   "    "
                ;
                ; NOTE: UPON DETECTION OF AN ERROR THE TEST WILL
                ;       RETRY THE OPERATION FOUR TIMES.
                ;
                ;*****************************************
                ;       6063/64 DISK STATUS WORD
                ;*****************************************
                ;       BIT     MEANING
                ;       0       ERROR
                ;       3       BUS ENABLE
                ;       4-5     DISK CAPACITY
                ;       6       IDLE DONE
                ;       7       WRITE PARITY
                ;       8       DCH ERROR
                ;       9       R/W TIMEOUT
                ;       10      DISK RDY
                ;       11      UNSAFE
                ;       12      DATA LATE
                ;       13      ECC ERROR
                ;       14      DATA VERIFY ERROR
                ;       15      PAGE DONE
```

```
!0017  .MAIN
;5.3.14 MOVING HEAD DISK TEST
;
;        AC0      GOOD DATA (SEE MH.SA)
;        AC1      BAD DATA
;        AC2      ADRS. OF GOOD DATA
;                 BAD IS AT AC2+4 IF MH.SA IS A -#
;        IN ADDITION THE FOLLOWING LOC'S ARE TYPED
;        MH.SA    =ADDRESS OF A RANDOM DATA CONTROL WORD
;                 (MHDW1 TO MHDW4) THE ERR WAS IN ONE OF
;                 THE FIRST 4 WORDS IN THE BUFFER
;                 =-# ERR IS AT AC2+4
;                 =AC2  ERROR WAS DISK STATUS
;                        IN WHICH CASE:
;                 AC2=DISK STATUS
;                 AC1=DIC DISK FOR MHDSK
;                 AC0=DIB DISK
;        MHDST    DATA START IN CORE
;        MHCST    DATA START FOR DCH MAP
;        MHSTA    LAST DISK STATUS
;        MHDOA    LAST DOA TO DISK
;        MHDOC    LAST DOC TO DISK
;
; ALSO IF THE ERROR OCCURS IN A READ OPERATION
; THE FOLLOWING DATA IS PRINTED:
;
;   WRITE MHCST = XXXXX
;        MD1      MD2      MD3      MD4
;        GGGG     HHHH     JJJJ     KKKK
; WHERE,
;        XXXXX = STARTING CHANNEL ADDRESS OF WRITE OPERATION
;         GGGG = FIRST PHYS 1K USED IN DISK WRITE
;         HHHH = 2ND     "     "   "    "    "    "
;         JJJJ = 3RD     "     "   "    "    "    "
;         KKKK = 4TH     "     "   "    "    "    "
;
; NOTE: UPON DETECTION OF AN ERROR THE TEST WILL
;        RETRY THE OPERATION FOUR TIMES.
;
;*************************************************
;        MOVING HEAD DISK STATUS WORDS
;*************************************************
;                                          6050,51
; BITS  4047            4048,57   4231     6030,45
;  0    R/A DONE          SAME     SAME     SAME
;  1    SEEK 0 DNE        SAME     SAME     SAME
;  2    SEEK 1 DNE        SAME     SAME     SAME
;  3    SEEK 2 DNE        SAME     SAME     SAME
;  4    SEEK 3 DNE        SAME     SAME     SAME
;  5    SEEK ON DRV 0     SAME    DUAL PRC DKT
;  6    SEEK ON DRV 1     SAME    SECT ERR V.S.
;  7    SEEK ON DRV 2     SAME    HEAD ERR N/A
;  8    SEEK ON DRV 3     SAME    ADDR ERR UNSAFE
;  9    DRIVE RDY         SAME     SAME     SAME
; 10    SEEK ERR          SAME     SAME     SAME
; 11    EOC ERR           SAME     SAME     SAME
; 12    UNSAFE          ADDR ERR UNSAFE ADDR ERR
; 13    ECC ERR           SAME     SAME     SAME
; 14    DATA LATE         SAME     SAME     SAME
; 15    ERR               SAME     SAME     SAME
;*************************************************
```

```
0018  .MAIN
;5.3.15 6060/61 DISK TEST
;
;        AC0      GOOD DATA (SEE ZB.SA)
;        AC1      BAD DATA
;        AC2      ADRS. OF GOOD DATA
;                 BAD IS AT AC2+4 IF ZB.SA IS A -#
;        IN ADDITION THE FOLLOWING LOC'S ARE TYPED
;        ZB.SA    =ADDRESS OF A RANDOM DATA CONTROL WORD
;                 (ZBDW1 TO ZBDW4) THE ERR WAS IN ONE OF
;                 THE FIRST 4 WORDS IN THE BUFFER
;                 =-# ERR IS AT AC2+4
;                 =AC2  ERROR WAS DISK STATUS
;                        IN WHICH CASE:
;                 AC2=DISK STATUS
;                 AC0=DIB DISK
;        ZBDST    DATA START IN CORE
;        ZBCST    DATA START FOR DCH MAP
;        ZBSTA    LAST DISK STATUS
;        ZBDOA    LAST DOA TO DISK
;        ZBDOC    LAST DOC TO DISK
;
; ALSO IF THE ERROR OCCURS IN A READ OPERATION
; THE FOLLOWING DATA IS PRINTED:
;
;   WRITE ZBCST = XXXXX
;        MD1      MD2      MD3      MD4
;        GGGG     HHHH     JJJJ     KKKK
;
; WHERE,
;        XXXXX = STARTING CHANNEL ADDRESS OF WRITE OPERATION
;         GGGG = FIRST PHYS 1K USED IN DISK WRITE
;         HHHH = 2ND     "     "   "    "    "    "
;         JJJJ = 3RD     "     "   "    "    "    "
;         KKKK = 4TH     "     "   "    "    "    "
;
; NOTE: UPON DETECTION OF AN ERROR THE TEST WILL
;        RETRY THE OPERATION FOUR TIMES.
;
;*************************************************
;        6060/61 STATUS WORDS
;*************************************************
; BITS  DIA              DIB
;  0    CNTL FULL        INVALID STATUS
;  1    R/A DONE         DRV RESERVED
;  2    SEEK 0 DONE      TRESPASSED
;  3    SEEK 1 DONE      READY
;  4    SEEK 2 DONE      BUSY
;  5    SEEK 3 DONE      OFFSET
;  6    PARITY ERROR     WRITE DISABLE
;  7    ILLEGAL SECT     N/A
;  8    ECC ERROR        ILLEGAL ADDR
;  9    BAD SECTOR       ILLEGAL CMD
; 10    CYL ERROR        PWR FAULT
; 11    SURF/SECT ERR    PACK UNSAFE
; 12    VERIFY ERROR     POSITIONER
; 13    R/A TIMEOUT      CLK FAULT
; 14    DATA LATE        WRITE FAULT
; 15    RD/WRT FAULT     DRIVE DONE
;*************************************************
```

```
!0019  .MAIN
                ;5.3.16 MAGNETIC TAPE TEST
                ;       AC0     GOOD DATA
                ;
                ;       AC1     BAD DATA
                ;       AC2     ADRS OF BAD DATA (GOOD IS AT AC2-4)
                ;IN ADDITION THE FOLLOWING LOC'S ARE TYPED
                ;       MODE    0=REWIND 1=WRITE 2=BACKSPACE OR READ
                ;               DRIVE # WILL APPEAR IN BITS 4,5&6
                ;       STATUS  LAST TAPE STATUS
                ;       MT.ER   ERROR COUNTER STARTS AT 6 AND COUNTS
                ;               DOWN FOR EACH REREAD
                ;       #READ   NUMBER OF BLOCKS READ
                ;       W/DOB   LOGICAL DCH ADDR USED IN WRITING TAPE
                ;     LAST/DOB  CURRENT DCH LOGICAL ADDRESS
                ;
                ;NOTE:  IF STATUS INDICATES TAPE ERR (BIT 0=1)
                ;       THE CONTENTS OF AC0,1,AND 2 SHOULD BE IGNORED**.
                ;
                ;*****************************************
                ;  TAPE STATUS WORD - DIA X,MTA
                ;*********************************************
                ;  BIT  MEANING          BIT     MEANING
                ;
                ;  0    ERROR            8       BEGINNING OF TAPE
                ;  1    DATA LATE        9       9 TRACK
                ;  2    REWINDING        10      BAD TAPE
                ;  3    ILLEGAL          11      SEND CLOCK
                ;  4    HI-DENSITY       12      FIRST CHARACTER
                ;  5    PARITY ERROR     13      WRITE LOCK
                ;  6    END OF TAPE      14      ODD CHARACTER
                ;  7    END OF FILE      15      UNIT READY
                ;*****************************************
```

```
!0020  .MAIN
                ;5.3.17 CASSETTE TAPE TEST
                ;
                ;       AC0     GOOD DATA
                ;       AC1     BAD DATA
                ;       AC2     ADRS OF BAD DATA (GOOD IS AT AC2-4)
                ;IN ADDITION THE FOLLOWING LOC'S ARE TYPED
                ;       MODE    0=REWIND 1=WRITE 2=BACKSPACE OR READ
                ;               DRIVE# WILL APPEAR IN BITS 4,5 & 6
                ;       STATUS  LAST TAPE STATUS
                ;       CA.ER   ERROR COUNTER STARTS AT 6 AND COUNTS
                ;               DOWN FOR EACH REREAD
                ;       #READ   NUMBER OF BLOCKS READ
                ;       W/DOB   LOGICAL DCH ADDR USED IN WRITING TAPE
                ;     LAST/DOB  CURRENT DCH LOGICAL ADDRESS
                ;
                ;NOTE:  IF STATUS INDICATES TAPE ERR (BIT 0=1)
                ;       THE CONTENTS OF AC0,1,AND 2 SHOULD BE IGNORED**.
                ;
                ;*****************************************
                ;  CASSETTE TAPE STATUS WORD - DIA X,CAS
                ;*********************************************
                ;  BIT  MEANING          BIT     MEANING
                ;
                ;  0    ERROR            8       BEGINNING OF TAPE
                ;  1    DATA LATE        9       =1
                ;  2    REWINDING        10      WRITE FAIL
                ;  3    ILLEGAL          11      0
                ;  4    =1               12      0
                ;  5    PARITY ERROR     13      WRITE LOCK
                ;  6    END OF TAPE      14      0
                ;  7    END OF FILE      15      UNIT READY
                ;*****************************************
                ;
                ;
                ;5.3.18 LINE PRINTER
                ;
                ;       NO ERROR TYPEOUTS.
                ;       PRINTER OUTPUT MUST BE EXAMINED VISUALLY.
                ;
```

```
;5.3.19 PROGRAMMABLE INTERVAL TIMER
;
;IF THE RTC AND PIT ARE RUN TOGETHER, THE PIT WILL MONITOR
;AND UPDATE THE RTC IF ANY DIFFERENCE EXISTS.  IF A DIFFERENCE
;OF MORE THAN 3 SECONDS IS FOUND IN ANY ONE MINUTE INTERVAL,
;THE FOLLOWING ERROR MESSAGE WILL BE PRINTED:
;        RTC AND PIT OUT OF SYNC XX SECONDS
;IF EITHER THE RTC OR THE PIT ARE DISABLED, THEN THE PIT TEST
;WILL GENERATE NO ERROR MESSAGES
;
```



```
;
;5.4     SPECIAL CASE ERROR TYPEOUTS
;
;5.4.1  POWER FAIL INTERRUPT
;        UPON DETECTION OF A POWER FAIL INTERRUPT
;        THE LOGICAL ADRS. OF THE P.C. AT INTERRUPT
;        WILL BE SAVED.
;        IF AUTO-RESTART  IS ENABLED OR THE POWER
;        FAIL WAS ONLY MOMENTARY, THE TEST WILL RE-
;        START AS IN A START AT 206 AFTER TYPING
;POWER FAIL @XXXXXX (WHERE XXXXXX IS THE PC AT INTR.)
;
;5.4.2  DATA CHANNEL PROTECT ERROR
;        IF BIT 0 OF THE MMPU DIC STATUS
;        REGISTER IS EVER DETECTED TO BE A 1,
;        THE TEST WILL ATTEMPT TO ACCUMULATE
;        AN INDICATION OF THE FREQUENCY OF THE
;        ERRORS.
;        PROG #  INDICATES THE TEST BEING EXECU-
;                TED WHEN THE DCH ERRORS WHERE
;                DETECTED.(MAY BE ANY TEST
;                AS DCH'S OCCUR IN THE BACKGROUND)
;        U ERRS  INDICATES THE NUMBER OF ERRORS
;                THAT COULD BE DETERMINED TO OCCUR
;                WITH THE MMPU IN "USER MODE"
;        S ERRS  INDICATES THE NUMBER OF ERRORS THAT
;                COULD BE DETERMINED TO OCCUR
;                WITH THE MMPU IN "SUPERVISOR MODE"
;
;NOTE: IT IS PROBABLY SAFE TO ASSUME THAT THE
;        PERIPHERAL THAT IS THE RECIPIENT OF THESE
;        DCH ERRORS WILL INTERJECT A FEW DATA
;        ERROR TYPEOUTS OF IT'S OWN.
```

```
;
;5.4.3  ILLEGAL SUPERVISOR CALL
;
;       THERE ARE SEVERAL CASES OF ILLEGAL SUPER-CALL'S
;       1.) IF THE LOCATION INDICATED IS 000001
;         THEN THE PROGRAM DETECTED THAT FOR SOME REASON
;         LOCATION 000000 WAS EXECUTED. THE ILLEGAL
;         SUPERCALL WAS FORCED BY THE ERROR HANDLER
;         TO INDICATE WHICH TEST WAS EXECUTING AT THE
;         TIME OF ERROR AND ITS ASSOCIATED INFORMATION.
;       2.) IF OTHER THAN LOCATION 000001 THEN
;         THE SUPERCALL HANDLER WAS ENTERED AND UPON
;         CHECKING THE LOCATION FOLLOWING THE SUPER-
;         CALL THERE WASN'T A VALID "JSR @XXXX".
;         THIS COULD HAPPEN IN SEVERAL WAYS:
;        A.) THE PROGRAM COULD HAVE JUMPED INTO
;          A DATA AREA WHICH CONTAINED A WORD
;          RECOGNIZED AS A SUPERCALL. IN THIS
;          CASE AC3 OF THE SECOND ENTRY OF THE
;          STACK USUALLY ISN'T 177510(SYC 3,3).
;        B.) THE JSR @XXXX WORD WAS OVER WRITTEN
;          BY ANOTHER TEST.
;        C.) THE 177510(SYC 3,3) OR THE JSR @XXXX
;          WAS IN A BAD MEMORY AREA CAUSING THE
;          WORDS TO DROP OR ADD BITS.
;        D.) THE JSR @XXXX WAS IN A DIFFERENT MAP
;          PAGE WHOSE CONTENNTS ISN'T CORRECT.
;        E.) SOME OTHER REASON ??
;
;  IN ANY CASE THE FOLLOWING INFORMATION IS TYPED:
;       THE STACK SHOULD CONTAIN TWO BLOCKS.
;       THE FIRST IS PUSHED BY THE ILLEGAL CALL.
;        (IF SYC 3,3 THEN HEADER  IS CORRECT)
;       THE SECOND IS BUILT BY THE ILLEGAL CALL
;       HANDLER AND INCLUDES THE FOLLOWING INFO.
;       AC0      ILLEGAL CALL IN LOC. XXXXX
;       AC3      CONTENTS OF LOC. XXXXX-1
;                (SHOULD = SYC 3,3 OR 177510)
```



```
;
;5.4.4  ERROR TRAP
;       AN I/O,WRITE,DEFER OR VALIDITY TRAP
;       OCCURED THAT WAS NOT FORCED BY ANY TEST
;       AND DID NOT = A READS 0 INSTRUCTION
;       THE STATE BLOCK TYPED IS THAT PUSHED AS A
;       RESULT OF THE TRAP
;       IN ADDITION THE FOLLOWING ARE TYPED:
;DIA MAP        CONTENTS OF THE MAP STATUS "A"
;                (USER-LEF-WP,ETC.)
;DOB             THE LAST BLOCK ALWAYS MAPPED FEATURE
;                IS UTILIZED TO RETRIEVE THE INSTR.
;                THAT TRAPPED.
;
;5.4.4.1 OLD MAP DESCRIPTION
;DIC             MAP STATUS "C" AS FOLLOWS
;        BIT 0   DCH PROTECT
;        BIT 1   SINGLE REFERENCE TRAP
;        BIT 2   WRITE PROTECT TRAP
;        BIT 3   VALIDITY PROTECT TRAP
;        BIT 4   DEFER VIOLATION
;        BIT 5   I/O PROTECT
;        BIT 6   USER A/B
;        BIT 7   P.C. UNDEFINED
;        BITS 9-15 PHYSICAL BLK #
;                  (377=VALIDTY PROTECT)
;
;
; 5.4.4.2 NEW MAP DESCRIPTION
; DIA             MAP STATUS AS FOLLOWS
;        BIT 1   EXT. FAULT
;        BIT 2   I/O PROTECT TRAP
;        BIT 3   WRITE PROTECT TRAP
;        BIT 4   DEFER PROTECT TRAP
;        BIT 5   SINGLE REF. TRAP
;        BIT 6-8 FORMAT BITS
;        BIT 9   LEF MODE
;        BIT 10  I/O PROTECT ENABLE
;        BIT 11  WRITE PROTECT ENABLE
;        BIT 12  DEFER PROTECT ENABLE
;        BIT 13  A/B USER SEL.
;        BIT 14  DCH ENABLE
;        BIT 15  USER MODE
;
; DIC    MAP STATUS "C" AS FOLLOWS
;        BIT 0   WRITE PROTECT BIT
;        BIT 1-3 FORMAT BITS
;        BIT 5   EXT PAGE
;        BITS 6-15 PHYSICAL BLK #
;                  (1777=VALIDTY PROTECT)
;
;5.4.5  INTERRUPT WAIT ELAPSED
;       THE PERIPHERAL DEVICE ASSOCIATED WITH THE
;       PROG. NUMBER TYPED HAS NOT RESPONDED WITH
;       A PROGRAM INTERRUPT FOR AN EXTENDED
;       PERIOD OF TIME. THE 2ND NUMBER TYPED
;       SHOULD POINT AT THE INTERRUPT HANDLER
;       FOR THE DEVICE THAT FAILED
```

```
;
;5.4.6  ERROR CORRECTION INTERRUPTS
;       IF ERCC IS ENABLED THE TYPE OUTS THAT
;       OCCUR WILL MORE LOGICALLY BE DUE TO
;       ERROR CORRECTION INTERRUPTS THAN
;       TO DATA FAILURES DETECTED BY
;       ANY OF THE GOOD/BAD COMPARES
;       AGAIN, THE INITIAL TYPEOUT WILL CONSIST
;       OF THE FOUR AC'S:
;       AC0= DATA WORD AFTER CORRECTION
;
;    IF OLD MAP(MMPU):
;
;       AC1= DIB FROM ERCC OPTION
;               BITS 0 TO 4 ARE CORRECTION CODE
;               BIT 15 INDICATES LOWER 64K IF=0
;                      UPPER 64K IF=1
;       AC2= PHYSICAL ADDRESS OF THE ERROR
;               (WITHIN 64K INDICATED)
;
;    IF NEW MAP(MMPU1):
;
;       AC1= DIB FROM ERCC
;         BITS 0-4 ARE CORRECTION CODE
;         BITS 13-15 ARE THE COMPLEMENT
;               OF HIGH ORDER PHYS ADDR BITS
;       AC2= DIA FROM ERCC
;         BITS 0-3 ARE COMPLEMENT OF PHYS
;               ADDR BITS XPA2,PA1-3
;         BITS 12-15 ARE THE COMPLEMENT
;               OF PHYS ADDR BITS PA12-15
;
;       AC3= ADDRESS(PC) OF THE INTERRUPT
;               (REFER TO LISTING)
;       THE CORRECTION CODE IS RIGHT JUSTIFIED
;       AND TYPED
;       THE TOTAL NUMBER OF ERROR CORRECTION
;       INTERRUPTS THAT OCCURED DURING THE LAST
;       TEST LOOP ARE TYPED
```



```
;
;FOR CONVENIENCE THE FOLLOWING TABLE
;IS INCLUDED TO HELP DETERMINE WHICH BIT
;IS FAILING IN AN ERROR CORRECTION MEMORY
;COR.CODE        BIT
;  0             NO ERROR
;  1             CHECK BIT 4
;  2             CHECK BIT 3
;  3             DATA BIT 0
;  4             CHECK BIT 2
;  5             DATA BIT 1
;  6             MULTIPLE BIT
;  7             DATA BIT 3
;  10            CHECK BIT 1
;  11            DATA BIT 4
;  12            ALL 21 BITS WERE=1
;  13            DATA BIT 6
;  14            DATA BIT 7
;  15            DATA BIT 8
;  16            DATA BIT 9
;  17            MULT. BITS FAILED
;  20            CHECK BIT 0
;  21            DATA BIT 11
;  22            DATA BIT 12
;  23            DATA BIT 13
;  24            DATA BIT 14
;  25            ALL 21 BITS READ AS 0'S
;  26            DATA BIT 2
;  27            MULTIPLE BIT'S
;  30            DATA BIT 10
;  31            MULTIPLE BITS
;  32            DATA BIT 5
;  33            MULTIPLE BITS
;  34            DATA BIT 15
;  35            MULTIPLE BITS
;  36            SAME
;  37            SAME
;
;THE DATA TYPED BY THE ERROR CORRECTION
;TYPEOUT IS AFTER CORRECTION HAS OCCURED.
;THEREFORE, FOR THE BIT TO HAVE FAILED,
;IT WOULD HAVE BEEN READ AS THE COMPLIMENT
;OF HOW IT APPEARS IN THE TYPEOUT
;
;NOTE THAT ALL SINGLE BIT ERROR CODES
;    INDICATE A CORRECTION BIT FAILED
```

```
;
;6.2    PROGRAM INITIALIZE
;THE DIAGNOSTIC LINKER INITIALIZES ITSELF
;AND INDIVIDUAL TESTS IN THE FOLLOWING
;SEQUENCE:
;   1. SYSTEM IS RESET, MAP OPTION IS
;      DETERMINED TO EXIST OR NOT EXIST
;      AND SWITCHES ARE SET UP
;      ACCORDINGLY
;   2. ANY OTHER NECESSARY CONSTANTS
;        ARE INITIALIZED
;      (MEM ALLOCATION TABLES)
;   3. INTERRUPT VECTOR TABLES ARE SET UP TO
;      PROCESS UNEXPECTED DEVICE INTERRUPTS
;   4. MEMORY IS SIZED IN 1K INCREMENTS
;      FROM 0 TO 128K AND BUILDS AN 8 WORD
;      BIT MAP OF EXISTING CONTIGUOUS
;      MEMORY(256K-16 WORDS FOR MMPU1).
;
;   5. THE EXIST MAP IS MOVED TO THE
;      AVAILABLE MAP AND EACH BIT
;      CORRESPONDING TO 1K OF UITILIZED
;      MEMORY IS REMOVED FROM THE MAP
;      SO THAT IT WILL NOT BE ASSIGNED
;      AS A SCRATCH AREA TO ANY TEST.
;      (INCLUDES PROGRAM STORAGE, MEMORY ALLOC.
;      TABLES, INTERRUPT MASKS AND STACK AREA AND
;      THE LAST 1K OF MEMORY TO PRESERVE THE
;      BINARY LOADER)
;   6. EACH TEST IS ENTERED IN SEQUENCE AT ITS
;      INIT. ENTRY POINT. OPTION TESTS DETERMINE
;      IF THE DEVICE THEY ARE ASSOC. WITH EXISTS
;      OR NOT AND PASS INTERRUPT SERVICE PARAM'S
;      TO THE LINKER.
;        (DEV#, MASK AND INTERRUPT SERVICE
;      ADDRESS)
;   7. LINKER THEN TYPES THE SYSTEM SIZE
;      INFORMATION ALONG WITH THE PROGRAM
;      RUN LIST AND WILL ALLOW THE OPERATOR
;      TO SELECT OR DELETE SPECIFIC TESTS
;       AND ENTER KEY OPTIONS
;      IF START WAS 00202 OR 204.
;   8. AFTER STARTING, THOSE TESTS THAT HAVE
;   "SIZED" THEIR SUBSYSTEM FOR SPECIFIC
;   PARAMETERS TYPE AN INDICATION OF THE PARAMETERS
;   THEY DETERMINED TO EXIST.(SEE THE
;   INDIVIDUAL DISK TEST DESCRIPTIONS.)
```



```
;6.3    PROGRAM RUN
;       ONCE THE LINKER HAS COMPLETED ALL
;       INITIALIZATION THE FOLLOWING SERIES
;       OF OPERATIONS IS LOOPED THROUGH
;
;          1. LINKER RANDOMLY SELECTS ONE OF
;             THE INDIVIDUAL TESTS UNTIL IT
;             FINDS ONE THAT IS NOT WAITING
;             FOR INTERRUPT (WAIT IS BIT 0 OF
;             THE THIRD WORD IN TEST=1) AND THAT
;             THE NEXT RANDOM NUMBER FALLS WITHIN
;             ITS ENTER LIMITS
;
;          2. IF THE MAP OPTION EXISTS, ALL LOGICAL PAGES
;             EXCEPT PAGE 0 ARE ACCESS PROTECTED WITH
;             THE PHYSICAL AREA OF THE SELECTED TEST
;             MAPPED TO ITSELF AND ANY ASSIGNED
;             SCRATCH AREA MAPPED TO START AT 1K
;             ABOVE THE TEST, MEMORY LOCATIONS SCRLO
;             AND SCRHI (SCRATCH LOW AND HIGH) ARE
;             SET TO INDICATE THE LIMITS OF
;             THE SCRATCH AREA AVAILABLE TO THE TEST.
;
;          3. DATA CHANNEL LIMITS (DCHLO AND DCHHI)
;             ARE CALCULATED AND ENTERED
;
;          4. THE SELECTED TEST IS ENTERED AT
;             ITS SPECIFIED EXECUTE ENTRY POINT
;
```

```
;6.4     INDIVIDUAL TEST DESCRIPTIONS
;
;6.4.1  ERROR CORRECTION
;
;  THE ERROR CORRECTION OPTION IN MULTIPROGRAMMING
; NORMALLY ENABLES ERROR CORRECTION IN MODE 3 (CORRECT
;AND INTERRUPT).  IF AN ERROR CORRECTION INTERRUPT OCCURS THIS
;TEST SAVES ALL INFORMATION AVAILABLE PERTAINING TO THE FIRST
;ERROR FOR LATER TYPEOUT. BETWEEN THE TIME OF THE ERROR CORRECTION
;INTERRUPT AND COMPLETION OF THE TYPEOUT ERROR CORRECTION IS
;RE-ENABLED IN "MODE 3". HOWEVER, UNTIL THE TYPEOUT IS COMPLETED,
;ERROR CORRECTION INTERRUPTS ARE ONLY COUNTED
;NOTE:  THE TYPEOUT DOES NOT OCCUR IMMEDIATELLY AT THE TIME
;       OF INTERRUPT, IT IS INITIATED AFTER
;       COMPLETION OF THE CURRENT PASS OF THE TEST THAT
;       WAS RUNNING WHEN THE INTERRUPT OCCURRED. ALSO,
;       IF THE ADDRESS OF THE ERROR WAS OUTSIDE OF THE
;       SCRATCH AND RESIDENT AREAS OF THE TEST THAT WAS
;       RUNNING, ONE SHOULD SUSPECT THAT THE ERROR
;       COULD HAVE OCCURRED DURING A DATA CHANNEL.
;
;6.4.2  CHECKERBOARD RAM
;
;THIS MEMORY CHECKER BOARD TEST IS A SUBSET OF OTHER MEMORY
;CHECKERBOARDS.  A COMPLETE TEST OF AN AVAILABLE SCRATCH
;AREA IS COMPRISED OF THE FOLLOWING SEQUENCE:
;
;CB.TK=0          ;REQUEST 1 TO 32K OF SCRATCH, RANDOMLY RE-
;                 ;LOCATE THE EXECUTE PORTION OF CHECKERBOARD
;                 ;INTO SCRATCH AND GENERATE THE CHECKERBOARD
;                 ;PATTERN
;
;CB.TK=1          ;DISTURB PASS-COMPLIMENT A SINGLE BIT IN EACH
;                 ;OF THE FIRST 16 WORDS OF SCRATCH, SHUFFLE THESE
;                 ;WORDS 16 TIMES SUCH THAT THEY END UP IN THEIR
;                 ;ORIGINAL POSITION, RE-COMPLIMENT THE SINGLE
;                 ;BIT IN EACH WORD-PROCEED WITH EACH GROUP OF
;                 ;16 WORDS UNTIL ALL MEMORY HAS BEEN EXERCISED.
;
;CB.TK=2          ;CHECK PASS-COMPARE EACH WORD IN SCRATCH WITH
;                 ;THE PATTERN EXPECTED
;
;CB.TK=3          ;FAST CHECKSUM MEMORY TO ENSURE THAT ALL DATA
;                 ;IS INTACT (RETURNS TO CHECK PASS IF CHECK-
;                 ;SUM DOES NOT AGREE.)
```



```
;6.4.3  EIS/MRI TEST
;
;THE MULTIPROGRAMMING EXTENDED INSTRUCTION SET - MEMORY
;REFERENCE INSTRUCTION TEST IS ESSENTIALLY ANOTHER MEMORY
;CHECKERBOARD EXERCISER.  THIS TEST, HOWEVER,INCLUDES 19
;BIT PATTERNS FOR ERROR CORRECTION MEMORY'S AND UTILIZES
;BYTE, BIT AND BLM INSTRUCTIONS TO EXERCISE THE MEM-
;ORY.  A COMPLETE PASS OF THE EIS/MRI TEST INCLUDES ALL
;OF THE FOLLOWING:
;
;EI.TK=0          ;ASSIGN SCRATCH (1 TO 32K), RANDOMLY RE-
;                 ;LOCATE EXECUTABLE CODE INTO THE SCRATCH AREA,
;                 ;GENERATE CHECKERBOARD PATTERN.(1 OF 4 RANDOM
;                 ;SELECTED IF ERROR CORRECTION)
;
;EI.TK=1          ;VERIFY THAT THE SCRATCH AREA CONTAINS THE
;                 ;CORRECT CHECKERBOARD PATTERN
;
;EI.TK=2          ;RANDOMLY SELECT GROUPS OF 16 WORDS COMLIMENT
;                 ;A SINGLE BIT IN EACH WORD, SHUFFLE 16 WORDS
;                 ;16 TIMES, RECOMPLIMENT THE SINGLE BIT IN
;                 ;EACH WORD.
;
;EI.TK=3          ;(SEE EI.TK=1) COMPARE
;
;EI.TK=4          ;SELECT RANDOM X DRIVERS, COMPLIMENT A
;                 ;SINGLE BIT ON EACH OF 16 DRIVERS (EVERY 64TH WORD)
;                 ;WORD) BLM ALL OF THE PATTERN AREA BACK TO ITSELF,
;                 ;RECOMPLIMENT BITS
;EI.TK=5                  ;(SEE EI.TK=1) COMPARE
;EI.TK=6-7-10     ;BLM THE PATTERN AREA TO ITSELF
;EI.TK=11         ;(SEE EI.TK=1) COMPARE
;EI.TK=12         ;BYTE TEST - LDB - COMPLIMENT
;                 ;STB LDB - RE-COMPLIMENT - STB EACH
;                 ;SEQUENTIAL BYTE IN THE PATTERN AREA.
;EI.TK=13         ;(COMPARE AGAIN)
;EI.TK=14         ;GENERATE COMPLIMENT WORST CASE PATTERN
;EI.TK=15         ;(SEE EI.TK=1) COMPARE
;EI.TK=16         ;(SEE EI.TK=2) BIT TEST
;EI.TK=17         ;(SEE EI.TK=3) COMPARE
;EI.TK=20         ;(SEE EI.TK=4) DRIVERS
;EI.TK=21         ;(SEE EI.TK=5) COMPARE
;EI.TK=22,23,23   ;(SEE EI.TK=6,7,10) BLM'S
;EI.TK=25         ;(SEE EI.TK=11) COMPARE
;EI.TK=26         ;(SEE EI.TK=12) BYTE'S
;EI.TK=27         ;(SEE EI.TK=13) COMPARE
```

```
;6.4.4  ARITHMETIC TEST
;
;THE MULTIPROGRAMMING RELIABILITY ARITHMETIC TEST WAS
;DERIVED FROM THE STAND ALONE ARITHMETIC TEST.  THIS TEST
;REQUIRES 2K OF SCRATCH FOR EXECUTION.  THE EXECUTE POR-
;TION OF THE TEST IS RANDOMLY RELOCATED WITHIN AVAILABLE
;SCRATCH.  IF THE SYSTEM IS MAPPED, (HAS AN MMPV) THE
;SCRATCH AREA IS RANDOMLY REMAPPED TO SOME OTHER LOGICAL AD-
;DRESS FOR EXECUTION.  AT THE END OF EACH EXECUTION PASS SCRATCH
;AREA IS RANDOMLY RELEASED OR HELD.  IF HELD, THE NEXT TIME
;THE TEST IS ENTERED, THE EXECUTABLE PORTION OF THE TEST WILL
;AGAIN BE RANDOMLY RELOCATED WITHIN SCRATCH FOR EXECUTION.
;
```



```
;6.4.5  ADDRESS TEST
;
;THE MULTIPROGRAMMING ADDRESS TEST GENERATES A VARIETY OF
;4 ADDRESS AND COMPLIMENT ADDRESS PATTERNS IN AVAILABLE
;SCRATCH.  THE TEST SEQ FOLLOWS:
;
;A.TSK=0          ;SELECT A SCRATCH AREA TO WORK IN, MOVE THE
;                 ;EXECUTE PORTION OF THE ADDRESS TEST TO
;                 ;EITHER THE HIGH OR LOW END OF SCRATCH
;                 ;GENERATE THE ADDRESS TO ADDRESS PAT-
;                 ;TERN IN THE UPWARD (LOW SCRATCH TO HIGH
;                 ;SCRATCH) DIRECTION, THE PATTERN IS GEN-
;                 ;ERATED IN TWO STEPS.  FIRST A RANDOM # OF
;                 ;WORDS (OFFSET A.AOS) IS FILLED VIA A BAM
;                 ;(ADRS(AC2))+1(AC0) TO ADRS(AC3) THE
;                 ;REST OF SCRATCH IS FILLED VIA A SECOND BAM
;                 ;(ADRS (AC2))+(OFFSET+1(AC0) TO ADRS AC3.
;
;A.TSK=1          ;VERIFY THAT EACH SCRATCH LOCATION CONTAINS
;                 ;ITS OWN LOGICAL ADDRESS.
;
;A.TSK=2          ;PATTERN GENERATION IS PERFORMED AS IN
;                 ;A.TSK=0 EXCEPT THAT THE 2 BAM'S FILL EACH
;                 ;SCRATCH LOCATION WITH THE 1'S COMPLIMENT OF
;                 ;ITS LOGICAL ADDRESS.
;
;A.TSK=3          ;VERIFY THAT EACH LOCATION CONTAINS THE 1'S
;                 ;COMPLIMENT OF ITS LOGICAL ADDRESS.
;
;A.TSK=4          ;EACH SCRATCH LOCATION IS AGAIN FILLED WITH
;                 ;ITS OWN LOGICAL ADDRESS.  HOWEVER, THE PATTERN IS
;                 ;GENERATED IN THE DOWNWARD DIRECTION (HIGH SCRATCH
;                 ;TO LOW SCRATCH) TWO STEPS ARE TAKEN.  A RANDOM
;                 ;# OF WORDS EQUAL TO OFFSET (A.AOS) ARE FILLED
;                 ;WITH THEIR LOGICAL ADRS.  THE REST OF SCRATCH IS
;                 ;THEN FILLED VIA AN ELDA AC2+OFFSET, OFFSET IS
;                 ;SUBTRACTED AND THE AN STH INTO THE LOWER
;                 ;LOCATION,
;
;A.TSK=5          ;(SEE A.TSK=1)
;
;A.TSK=6          ;GENERATION IS AS A.TSK=4 EXCEPT EACH LOCATION
;                 ;IS FILLED WITH THE 1'S COMPLIMENT OF ITS
;                 ;LOGICAL ADDRESS
;
;A.TSK=7                  ;(SEE A.TSK=3)
;
;A.TSK=10         ;THE PATTERN GENERATION OF A.TSK=0 IS RERUN
;
;A.TSK=11         ;(SEE A.TSK=1)
```

```
;6.4.6  EXTENDED ADDRESSING TEST
;THE EXTENDED ADDRESSING TEST
;VERIFIES THE CORRECT OPERATION OF THE DOUBLE LENGTH
;INSTRUCTIONS,THE IMMEDIATE MODE DOUBLE LENGTH,
;DISPA,CLM,MSP,HLV AND ELEF.
;SCRATCH AREA IS TREATED AS INTHE EIS/MRI TEST
;WITH TEST EXECUTION SIMILAR TO THE ARITH,FLT PT.
;AND LEF/ERROR TESTS
;6.4.7  FLOATING POINT TEST
;
;THE MULTIPROGRAMMING FLOATING POINT TEST IS SIMILAR IN OP-
;ERATION TO THE ARITHMETIC TEST.  THE FPU TEST DATA BUFFER IS
;LOCATED RANDOMLY WITHIN 32 WORDS IN THE CENTER OF THE EX-
;ECUTE PORTION OF THE FLT PT TEST.  THE FLT. PT. NUMBERS PRO-
;CCESSED BY THIS TEST ARE FIXED (NOT RANDOM) AND CAN BE FOUND
;SPECIFIED ON THE LISTING FOR EACH TEST.
;
;6.4.8  LEF/ERROR TEST
;
;THE LEF MODE - ERROR TEST IN MULTIPROGRAMMING RELIABILITY IS
;ONLY RUN IF AN MMPU EXISTS.  THE LEF PORTION OF THIS TEST VERIFIES
;THAT THE LEF ENABLE ON THE MMPU FUNCTIONS IN ALL ADDRESSING
;MODES.  THE ERROR PORTION OF THE TEST VERIFIES THAT THE WRITE, I/O
;DEFER AND VALIDITY PROTECT FEATURES OF THE MMPU FUNCTION
;CORRECTLY.  SCRATCH AND EXECUTION ARE TREATED AS IN ARITH-
;METIC TEST.
;
```



```
;6.4.9  COMMERCIAL INSTRUCTION TEST
;
;UPON ENTERING FOR INITIALIZATION THIS TEST DOES A TRIAL
;INSTRUCTION TO DETERMINE IF THE COMMERCIAL OPTION AND
;THE FLOATING POINT OPTION ARE INSTALLED TOGETHER.
;UPON ENTERING FOR EXECUTION THE TEST TRIES TO ACQUIRE
;4K OF SCRATCH. IF OBTAINED THE TEST MODULE IS
;MOVED UP INTO SCRATCH LEAVING THE AREA ABOVE AND
;BELOW THE TEST AS DATA SCRATCH BUFFERS.
;THE SIX INSTRUCTIONS TESTED ARE: ELDB,ESTB,CMV,CMP
;CTR, AND CMT.
; THE EXTENDED LOAD BYTE,STORE BYTE INSTRUCTIONS ARE
;TESTED TOGETHER. A RANDOM BYTE ADDRESS IS GENERATED IN
;EITHER THE HIGH OR THE LOW SCRATCH BUFFER. A RANDOM
;NUMBER (16 BITS) IS THEN STORED INTO MEMORY LOCATION
;CONTAINING THAT BYTE. BITS 8-15 OF THIS RANDOM NUMBER
;IS AGAIN STORED INTO THAT BYTE ADDRESS BY THE ESTB INSTR.
; THESE TWO ADJACENT BYTES OF THE SAME WORD ARE THEN LOADED
;ONE AT A TIME BY THE ELDB INSTRUCTION BACK INTO THE ACCUM.
;THE RANDOM NUMBER IS HENCE RECONSTRUCTED AND COMPARED WITH
;THE ORIGINAL FOR ERROR. THIS EXERCISE CHECKS THAT THE ELDB,
; ESTB INSTRUCTIONS DO NOT DISTURB THE ADJACENT BYTES.
;ALL FOUR ADDRESSING MODES ARE TESTED SEPERATELY.
;  THE CMV, CMP, CTR, AND CMT INSTRUCTIONS ALL DEAL
;WITH STRINGS OF BYTES FROM A SOURCE TO A DESTINATION
;FIELD. THE SIZES OF THE LOW AND HIGH BUFFER ARE COMPARED
;AND THE SMALLER BUFFER IS SELECTED AS THE SOURCE
;BUFFER WHILE THE LARGER IS THE DESTINATION. THE SOURCE
;BUFFER IS FILLED WITH RANDOM DATA AT THE START OF THE
;TEST IN ORDER TO MINIMIZE THE USE OF THE RANDOM DATA
;GENERATOR. THE CMV AND CMP INSTRUCTIONS ARE EXERCISED TOGETHER.
;THE CTR AND CMT INSTR. ARE EXERCISED SEPERATELY
;AND THEIR TRANSLATION TABLE IS LOCATED RANDOMLY IN THE SOURCE
;BUFFER. SINCE THE CTR INSTRUCTION REQUIRES A TRANSLATION TBL.
;OF 128 WORDS, A SCHEME IS SETUP WHICH DIVIDES THE
;LARGER BUFFER INTO TWO HALVES IF THE SMALLER BUFFER
;SIZE IS LESS THAN 256 WORDS. IN THIS CASE, THE LOWER HALF IS
;CHOSEN AS THE SOURCE AND THE UPPER HALF AS THE DESTINATION
;BUFFER.
; THE LDI,STI,LDIX,LSN AND FINT INSTRUCTIONS HANDLE THE
;CONVERSION OF INTEGERS AND FLOATING POINT NUMBERS.
;A RANDOM NUMBER IS FLOATED FROM MEMORY INTTO FPAC0,
;THE EXPONENT OF WHICH IS THEN RANDOMIZED. FPAC0 IS
;INTEGERIZED (BY FINT) AND STORED (BY STI) AS AN
;INTEGER 16 BYTES LONG. THIS INTEGER IS THEN LOADED BY
;LDI INTO FPAC1, AND COMPARED WITH FPAC0 FOR ERROR.
;THE SAME INTEGER IS ALSO TESTED BY THE LSN INSTRUCTION
;AND THE RETURNED CODE IN AC1 IS EXAMINED FOR ERROR.
;A SIMILAR APPROACH IS USED IN TESTING THE LDIX ,STIX
;INSTRUCTIONS. ALL EIGHT FORMATS OF INTEGERS(TYPES 0-7)
;ARE TESTED SEPERATELY.
; THE EDIT INSTRUCTION IS USED TO CONVERT INTERGERS
;FROM ONE FORMAT TO THE OTHER. THREE TESTS ARE
;WRITTEN IN WHICH ALL EDIT OP CODES ARE EXERCISED.
```

```
10035  .MAIN
01              ;6.4.10 DCU TEST
02              ;
03              ;THE MULTI-PROGRAMMING DCU TEST RUNS
04              ;AN ARITHMETIC TEST VIA THE DATA CHANNEL INTO
05              ;THE HOST MEMORY.
06              ;THE DCU INTERRUPTS THE HOST CPU WHEN EITHER IT
07              ;COMPLETES THE TEST OR UPON DETECTING AN ERROR.
08              ;
09              ;THIS TEST WILL AUTO-SIZE FOR THE EXISTANCE OF THE DCU AT
10              ;DEVICE CODE 64. IF OTHER DEVICE CODES ARE TO BE USED,
11              ;PATCH THEM INTO LOCATION DCUDV. IF NOT FOUND THE TEST
12              ;WILL BE AUTOMATICALLY DELETED.
13              ;6.4.11 SC MEMORY TEST
14              ;
15              ;THIS MEMORY TEST DOES A READ/MODIFY/WRITE TO THE AVAILABLE
16              ;SCRATCH AREA USING THE "ISZ AND DSZ" INSTRUCTION.
17              ;THE ISZ/DSZ TEST IS MOVED TO SCRATCH STARTING AT SCRLO
18              ;AND TESTED AREA STARTS AT FIRST LOCATION
19              ;ABOVE THE TEST. TEST IS BROKEN INTO THE
20              ;FOLLOWING CHECKS:
21              ;
22              ;       MM.TK= 0        WRITE INTO EACH MEMORY LOCATION A MINUS
23              ;                       ONE STARTING AT END OF TEST AND ENDING AT
24              ;                       SCRHI VERIFYING EACH GOT THERE.
25              ;
26              ;       MM.TK= 1        READ A LOCATION BEFORE DOING THE ISZ
27              ;                       TO VERIFY IT HASN'T BEEN DISTURBED.
28              ;
29              ;       MM.TK= 2        ISZ DIDN'T SKIP
30              ;
31              ;       MM.TK= 3        LOCATION NOT 0 AFTER ISZ
32              ;
33              ;       MM.TK= 4        DSZ SKIPPED-ERROR
34              ;
35              ;       MM.TK= 5        DSZ TST- LOCATION NOT -1 AFTER DSZ
36              ;
37              ;       MM.TK= 6        SAME AS 1, EXCEPT TESTING IN THE REVERSE
38              ;                       DIRECTION
39              ;
40              ;       MM.TK= 7        SAME AS 2, EXCEPT TESTING IN THE REVERSE
41              ;                       DIRECTION.
42              ;
43              ;       MM.TK= 10       SAME AS 3, EXCEPT TESTING IN THE REVERSE
44              ;                       DIRECTION.
45              ;
```

```
10036  .MAIN
01              ;6.4.12 WCS TEST
02              ;
03              ;       THE WCS TEST IS DIVIDED INTO THREE MAIN
04              ;       AREAS WHICH ARE DESCRIBED BELOW.
05              ;
06              ;6.4.12.1       DEC1,DEC2 TEST
07              ;
08              ;       A DEC1 IS PERFORMED TO A RANDOM LOCATION
09              ;       IN WCS, FROM THERE A DEC2 IS PERFORMED
10              ;       TO ANOTHER RANDOM LOCATION IN WCS AND THEN
11              ;       AN EXIT OCCURS. THE LAST THREE RANDOM
12              ;       NUMBERS ARE USED AS FOLLOWS:
13              ;
14              ;       AC0     BITS 8-15 = DEC1 ADDRESS IN WCS (0-377)
15              ;       AC1     BITS 8-15 = DEC2 ADDRESS IN WCS (0-377)
16              ;       AC2     BITS 12-15 = ENTRY # (0-17)
17              ;
18              ;6.4.12.2       ALU TEST
19              ;
20              ;       THE ALU TEST IS A GROUP OF MICRO-INSTRUCTIONS
21              ;       WHICH ARE LOADED WITH A RANDOM STARTING LOCATION
22              ;       IN WCS AND WHICH EXERCISE ALL THE ALU FUNCTIONS
23              ;       ALONG WITH FUNCTIONS IN THE FOLLOWING FIELDS:
24              ;       A-PORT,A-INPUT,B-PORT,SHIFT,LOAD, AND CARRY.
25              ;
26              ;       THE THREE RANDOM #'S IN AC0,AC1,AC2 ARE USED AS
27              ;       OPERANDS AND THE RESULTS FROM EXECUTION
28              ;       OF THE WCS SEQUENCE ARE COMPARED WITH SIMULATED
29              ;       RESULTS.
30              ;
31              ;       THE LAST THREE RANDOM NUMBERS ARE USED AS FOLLOWS:
32              ;       AC0     BITS 8-15       =FIRST ADDRESS IN WCS(0-377)
33              ;       AC1     BITS 12-15      =ENTRY # (0-17)
34              ;       AC0                     =OPERAND 1
35              ;       AC1                     =OPERAND 2
36              ;       AC2                     =OPERAND 3
37              ;
38              ;6.4.12.3       MEMORY TEST
39              ;
40              ;       THE WCS MEMORY TEST PRIMARILY EXERCISES THE
41              ;       MA AND MBUS FIELDS.
42              ;
43              ;       THE TEST SEQUENCE IN WCS GENERATES TWO
44              ;       NEW WORDS DEPENDENT UPON THE
45              ;       ORIGINAL RANDOM NUMBER IN TWO WORK LOCATIONS.
46              ;       THE MAIN PROGRAM SIMULATES THE PROCEDURE AND
47              ;       COMPARES THE ACTUAL WITH THE EXPECTED RESULTS.
48              ;
49              ;       THE LAST THREE RANDOM NUMBERS ARE USED AS FOLLOWS:
50              ;
51              ;       AC0     BITS 8-15       =FIRST ADDRESS IN WCS (0-377)
52              ;       AC1     BITS 12-15      =ENTRY # (0-17)
53              ;       AC0                     =ORIGINAL MR0
54              ;       AC1                     =ORIGINAL MR1
55              ;
```

```
10037  .MAIN
                ;6.4.13 NOVA DISK TEST
                ;
                ;WHEN ENTERED FOR ITS INITIALIZATION PASS THE
                ;NOVA DISK TEST SIZES THE DISK SYSTEM. IT DOES THIS BY
                ;INITIATING A READ AT THE HIGHEST ADDRESS OF
                ;EACH AVAILABLE DISK UNTIL IT NO LONGER RECEIVES A
                ;NONEXISTENT DISK BIT IN THE STATUS.
                ;THIS "HIGH DISK ADRS." IS TYPED THE FIRST
                ;TIME THE NOVA DISK TEST IS ENTERED.
                ;DURING RUNNING, THE DISK TEST KEEPS 3 RANDOM
                ;CONTROL TABLES THAT CONTAIN THE FOLLOWING INFO:
                ;        A. THE START SECTOR# OF 16 RANDOM SELECTED
                ;           SECTORS. (THIS # IS RIGHT JUSTIFIED 4 BITS)
                ;        B. A START SECTOR RANDOMLY SELECTED WITHIN 16
                ;        C. THE NUMBER OF SECTORS TO BE WRITTEN/READ
                ;        D. 4 RANDOM DATA WORDS THAT KEY THE PATTERN
                ;           (THESE 4 WORDS REPEAT EVERY 4 WORDS)
                ;THE TEST OPERATES OFF THESE RANDOM CONTROL TABLES
                ;IN A MANNER SIMILAR TO THAT DESCRIBED FOR THE INDIVIDUAL
                ;DISKS IN THE MOVING HEAD DISK TEST
                ;
```

```
10038  .MAIN
                ;6.4.14 6063/64  DISK
                ;
                ;DURING INITIALIZATION THE TEST CHECKS FOR THE
                ;EXISTANCE OF A DISK CONTROLLER AND THEN CHECKS
                ;FOR THE EXISTANCE OF ANY/ALL DRIVES. A WRITE
                ;BUFFERS COMMAND IS USED TO SIZE FOR AVAILABLE
                ;DRIVES.
                ;
                ;THE TESTING OF EACH AVAILABLE DRIVE IS CONTROLLED
                ;BY SELECTION OF ONE OF THREE OPERATION TABLES
                ;PER DRIVE. EACH CONTROL TABLE IS 13 WORDS IN
                ;LENGTH. THE FIRST WORD CONTAINS THE TRACK NUMBER
                ;(RANDOMLY SELECTED), THE SECOND WORD CONTAINS THE
                ;STARTING SECTOR AND NUMBER OF SECTORS USED. THE
                ;STARTING SECTOR IS RANDOMLY SELECTED AND THE NUMBER
                ;OF SECTORS IS DETERMINED BY THE AMOUNT OF SCRATCH
                ; AREA ASSIGNED TO THE DISK TEST WHEN THE DISK
                ;IS WRITTEN.
                ; THE THIRD WORD IS AN ERROR COUNTER FOR THE CURRENT
                ;CYLINDER AND IS USED IN RETRING ON ERROR.
                ;THE FOURTH WORD CONTAINS THE FIRST WORD OF THE
                ;COMMAND QUEUE,I.E.,DRIVE+TRACK+SECTOR.
                ;THE 5TH THRU 8TH WORDS ARE THE RANDOM DATA
                ;WORDS USED TO CREATE THE TEST PATTERN.(THEY REPEAT
                ;EVERY FOUR WORDS)
                ;THE 9TH WORD IS THE CHANNEL ADDR. USED IN
                ;WRITING TO THE DISK. WORDS TEN THRU 13
                ;ARE THE PHYSICAL MEMORY 1K'S USED TO WRITE
                ;TO THE DISK.
                ;
                ;UPON ENTERING FOR INITIAL EXECUTION, THE TEST ATTEMPS TO
                ;ACQUIRE 1-4K OF SCRATCH AREA. THE TEST THEN RANDOMLY
                ;SELECTS A DATA STARTING ADDRESS AFTER THE FIRST
                ;256 WORDS IN SCRATCH. THE FIRST 256 WORDS ARE RESERVED
                ;FOR THE COMMAND QUEUE.
                ;THE TEST THEN SELECTS ONE OF THE AVAILABLE
                ;DRIVES AND ONE OF THE THREE OP TABLES FOR THAT DRIVE.
                ;IF THE FIRST WORD OF THE TABLE IS NON-ZERO(INDICATING
                ;THE THE TRACK # IN WORD #1,STARTING AT THE SECTOR IN
                ;BITS 11-15 OF THE SECOND WORD,FOR THE # OF SECTORS
                ;SPECIFIED BY THE BITS 3-7 IN THE SECOND WORD, RANDOM DATA
                ;HAS BEEN WRITTEN THAT IS EQUAL TO THE CONTENTS OF WORDS
                ;4 THRU 7 F THE OP TABLE) THEN THE NEXT OPERATION OF
                ;READ OR DATA VERIFY IS RANDOMLY SELECTED.
                ;IF THE FIRST WORD OF THE TABLE IS ZERO, A TRACK IS
                ;SELECTED WHICH IS NOT CURRENTLY IN AN OP TABLE,
                ;AND A STARTING SECTOR # IS RANDOMLY SELECTED SUCH THAT
                ;THE # OF SECTORS WRITTEN WILL NOT MAKE THE SECTOR #
                ;FIELD OVERFLOW INTO THE TRACK FIELD.(I.E.,THE STARTING
                ;SECTOR FALLS BETWEEN 0 AND 32-# OF SECTORS TO BE WRITTEN)
                ;THE DATA PATTERN IS GENERATED IN SCRATCH AREA AND
                ;A WRITE OPERATION IS SELECTED.
                ;
                ;AFTER SELECTION OF THE OPERATION TO BE PERFORMED,
                ;A COMMAND QUEUE IS GENERATED IN THE FIRST 256 WORDS
                ;OF SCRATCH(FIVE WORDS PER SECTOR). THE FIRST WORD
                ;CONTAINS THE DRIVE,TRACK AND SECTOR TO BE EXERCISED.
```

```
!0039  .MAIN
                ;6.4.14 CONTINUED
                ;       THE SECOND WORD CONTAINS THE COMMAND AND EXTENDED
                ;       ADDRESS BITS OF THE DATA ADDRESS. WORD # 3 CONTAINS
                ;       THE THE LOWER 16 BITS OF THE LOGICAL ADDRESS OF THE
                ;       DATA . WORD # 4 IS USED BY THE DISK TO STORE THE
                ;       STATUS WORD. WORD # 5 IS NOT CURRENTLY USED.
                ;       THE LAST QUEUE BLOCK CONTAINS A HALT BIT IN WORD
                ;       # 2 WHICH TERMINATES THE OPERATION AFTER
                ;       COMPLETION OF THE SECTOR AND CAUSES THE DISK
                ;       TO INTERRUPT THE CPU.
                ;       THE DISK STATUS IS CHECKED AT THE TIME OF THE INTERRUPT
                ;       AND IF OK THE SCRATCH AREA IS VERIFIED. AS THE
                ;       DATA COMPARES ARE PERFORMED A NEGITIVE COUNT IS
                ;       STORED IN THE CHECKED LOCATION TO CLEAR THE BUFFER.
                ;
```

```
!0040  .MAIN
                ;6.4.15 MOVING HEAD DISK TEST
                ;
                ;WHEN ENTERED FOR INITIALIZATION, THE MOVING HEAD
                ;DISK TEST SIZES EACH DISK THAT IS "READY".
                ;A. THE HIGHEST AVAILABLE SECTOR IS FOUND BY A SERIES
                ;   OF SEEKS AND RECALIBRATE COMMANDS.
                ;B. THE NUMBER OF AVAILABLE SECTORS ON A SURFACE ARE
                ;   FOUND BY INITIATING A SERIES OF 2 SECTORS READS.
                ;C. THE NUMBER OF SURFACES ARE DETERMINED BY INITIATING
                ;   ANOTHER SERIES OF 2 SECTOR READS AT THE LAST
                ;   SECTOR ON EACH SURFACE UNTIL END OF CYLINDER.
                ;THIS INFORMATION IS TYPED (BY DISK) THE
                ;FIRST TIME THE TEST IS ENTERED DURING "RUN"
                ;NOTE:  BECAUSE OF A MORE COMPLEX METHOD OF
                ;       CONTROLLING MULTIPROCESSOR APPLICATIONS
                ;       THIS TEST CANNOT DETERMINE IF
                ;       THE DISK CONTROL IS ON THE PRIMARY OR
                ;       SECONDARY SIDE OF THE DISK ADAPTER
                ;THE TESTING OF EACH AVAILABLE DRIVE IS CONTROLLED BY
                ;AN INDIVIDUAL CONTROL TABLE FOR EACH DRIVE.
                ;THE OPERATION OF THE EACH MOVING HEAD DISK IS CONTROLLED BY
                ;THE CONTENTS OF 3 TABLES.  EACH CONTROL TABLE IS 13 WORDS
                ;IN LENGTH.  THE FIRST WORD CONTAINS THE CYLINDER NUMBER,
                ;(RANDOMLY SELECTED), THE SECOND WORD CONTAINS THE STARTING
                ;SECTOR AND NUMBER OF SECTORS UTILIZED.  THE START SECTOR IS
                ;RANDOMLY SELECTED AND THE NUMBER OF SECTORS IS CONTROLLED BY THE
                ;AMOUNT OF SCRATCH AREA AVAILABLE TO THE DISK TEST WHEN THE
                ;DISK IS WRITTEN.
                ;THE 3RD WORD IS AN ERROR COUNTER. FOR EACH ERROR DETECTED, THE
                ;DISK IS RECALIBRATED AND THE OPERATION IS REPEATED.
                ;THIS "RECAL/REPEAT" IS EXECUTED UP TO 4 TRYS.
                ;THE FOURTH THROUGH 7TH WORDS ARE THE RANDOM DATA
                ;USED TO GENERATE THE TEST PATTERN.(THEY REPEAT EVERY
                ;4 WORDS. THE 8TH WORD IS A RANDOM SEC.# THAT IS TREATED
                ;SINGLY AS THE CONTIGUOS SECTORS IN WORD 2. IF THE LAST
                ;SECTOR ON THE CYLINDER, READS AND WRITES FORCE THE EOC
                ;STATUS AS EVERY OPERATION IS DONE WITH A SEC. COUNT=2
                ;THE NINTH WORD IS THE CHANNEL ADDR. USED TO WRITE
                ;TO THE DISK. WORDS TEN THRU THIRTEEN ARE THE MEM. 1K'S
                ;USED IN WRITING TO THE DISK.
                ;
                ;WHEN INITIALLY ENTERED, THE DISK TEST ATTEMPTS TO ACQUIRE 1
                ;TO 4K OF SCRATCH AREA.  THE TEST THEN RANDOMLY SELECTS A DATA
                ;START ADDRESS WITHIN THE FIRST 256 WORDS OF SCRATCH.
                ;THE TEST THEN RANDOMLY SELECTS ONE OF THE AVAIL DISKS
                ;THE TEST THEN RANDOMLY SELECTS ONE OF THE 3 UP TABLES. IF
                ;THE FIRST WORD OF THE UP TABLE IS NOT=0 IT INDICATES THAT -
                ;AT THE CYLINDER NUMBER IN THE FIRST WORD - STARTING WITH THE
                ;SECTOR # IN BITS 6 TO 15 OF THE 2ND WORD - FOR THE # OF SECTORS
                ;SPECIFIED BY IN BITS 2 TO 5 OF THE 2ND WORD - RANDOM DATA HAS
                ;BEEN WRITTEN THAT IS EQUAL TO THE DATA IN WORDS 4 TO 7
                ;OF THE UP TABLE.  READ FROM DISK IS SELECTED.
                ;IF THE FIRST WORD OF THE TABLE IS=0 THE TEST - RANDOMLY
                ;SELECTS A CYLINDER NOT CURRENTLY IN AN UP TABLE - RANDOMLY
                ;SELECTS A START SECTOR (THE # OF SECTORS IS = TO THE AMOUNT
                ;OF SCRATCH AVAILABLE) - AND GETS FOUR RANDOM DATA WORDS WRITE
                ;TO DISK IS SELECTED
```

```
;6.4.15 (CONTINUED)
;
;THE TEST THEN INITIATES A SEEK TO THE CYLINDER SELECTED
;AND AT SUCCESSFUL COMPLETION OF THE SEEK EITHER READS OR
;WRITES THE # OF SECTORS AVAILABLE.
;STAUS ERROR OF DATA LATE WILL CAUSE THE TEST TO RETRY
;THE OPERATION UP TO FOUR TIMES.
;AT SUCCESSFUL COMPLETION OF EITHER THE READ OR WRITE, THE
;DATA BUFFER IS CHECKED TO VERIFY THAT IT CONTAINS THE
;CORRECT DATA.  AS DATA COMPARES CORRECTLY, THE CORRECT
;WORDS ARE FILLED WITH THE NEGATIVE COUNT=TO THE NUMBER
;OF WORDS LEFT IN THE BUFFER.
;
;6.4.16 6060/61 DISK TEST
;
;       THE 6060/61 DISK TEST IS SIMULAR TO THE
;       MOVING HEAD DISK TEST IN OPERATION.
;
;6.4.17 MAGNETIC TAPE OR CASSETTE TEST
;
;THE MULTIPROGRAMMING MAGNETIC TAPE AND CASSETTE TESTS ARE
;IDENTICAL IN OPERATION.  THREE TO SIXTY THREE RECORDS
;OF RANDOM DATA ARE WRITTEN, THE DATA BUFFER IS CHECKED,
;THE MAG TAPE OR CASSETTE IS BACKSPACED TO THE BEGIN-
;NING OF THE JUST WRITTEN RECORDS.  THEN, AS MANY RECORDS
;AS THE SCRATCH AREA WILL CONTAIN ARE READ BACK, THE
;DATA IS VERIFIED AND THE SEQUENCE (READ/VERIFY) IS RE-
;PEATED UNTIL ALL RECORDS IN THE SEQUENCE HAVE BEEN READ.
;
;FOR TAPE WRITE STATUS ERRORS, THE TAPE IS BACKSPACED/RE-
;WRITTEN UNTIL THE ERROR NO LONGER OCCURS.  FOR TAPE READ
;STATUS ERRORS THE TEST BACKSPACES AND REREADS A TOTAL OF
;3 TRYS.  STATUS ERRORS DURING BACKSPACE ARE CONSIDERED NON-
;RECOVERABLE.  FOR ALL ERRORS TYPED, THE TAPE IS REWOUND
;AND THE TEST RESTARTED AT LOAD POINT.
;
;WHEN THE TAPE REACHES EOT DURING THE WRITE OPERATION, TAPE
;IS REWOUND AND THE TEST RESTARTS AT LOAD POINT.
;
;THE RANDOM DATA IS A SEQUENCE OF 4 WORDS REPEATED EVERY 4TH
;WORD.  ALL RECORDS ARE 256 WORDS IN LENGTH.  AS DATA IS
;VERIFIED IN THE BUFFER IT IS REPLACED WITH A WORD EQUAL
;TO THE NEGATIVE COUNT OF THE NUMBER OF WORDS LEFT
;TO BE COMPARED.
;
;THESE TAPE TESTS UTILIZE 1 TO 6K OF SCRATCH AND THE
;DATA BUFFER START IS RANDOMLY SELECTED TO BE IN THE FIRST 256
;WORDS.
;
;ANY COMBINATION OF 1 TO 8 DRIVES MAY BE TESTED SIMPLY BY
;HAVING THEM ON LINE WRITE ENABLED.
```



```
;6.4.18 LINE PRINTER TEST
;
;THE NON-DCH LINE PRINTER TEST RANDOMLY PRINTS 10 TO 60 LINES
; PER PAGE WITH RANDOM STALLS EVERY 1 TO 9 LINES.
;EACH LINE OF PRINT CONSISIS OF THE CHARACTERS SPACE
;(40) TO Z (132).  THE TEST FILLS THE PRINT BUFFER UNTIL THE
;FIRST PRINT CYCLE STARTS.  CONTINUATION OF PRINTING UNTIL
;RANDOM STALL IS THEN RUN OFF INTERRUPTS FROM THE PRINTER.
;  THE DCH-LINE PRINTER TEST ASSIGNS 1 TO 2K OF
;SCRATCH AND ASSIGNS IT TO THE DCH A MAP. IT THEN RANDOMLY
;CHOSES A STARTING ADDRESS 0 TO 63 WORDS INTO THE SCRATCH AREA
; NEXT THE TEST CHOSES 10 TO A MAXIMUM OF 60 LINES TO PRINT.
;THE PATTERN PRINTED CONSISTS OF THE CHARACTERS SPACE(40) TO
; ](135). A TAB RUNAWAY ERROR WILL RESULT IN A PROGRAMED HALT.
;
;6.4.19 PROGRAMMABLE INTERVAL TIMER TEST
;
;THE PROGRAMMABLE INTERVAL TIMER IS SET FOR A 1 SECOND INTERVAL.
;ASSUMING THE PIT TO BE MORE RELIABLE THAN THE REAL TIME CLOCK,
;THE PIT TEST WILL UPDATE THE RTC EVERY MINUTE AND INDICATE AN
;ERROR IF THE TWO DIFFER BY MORE THAN 5% (3 SEC/MIN).  IF THE
;RTC DOES NOT EXIST, THE PIT ASSUMES THE REAL TIME CLOCK'S
;RESPONSIBILITY OF REPORTING THE TIME AT 5, 15, AND 30 MINUTES
;AND EVERY 30 MINUTES THEREAFTER. THE TIME WILL ALSO BE INDICATED
;AFTER EVERY ERROR.
;
;6.4.20 REAL TIME CLOCK
;
;THE REAL TIME CLOCK IS RUN AT 1K HERTZ.  RUNTIME ALONG
;WITH ACCUMULATED ERROR COUNT ARE PRINTED AT 5 MINUTES
;15 MINUTES, 30 MINUTES AND EVERY 30  MINUTES OF RUNTIME
;THEREAFTER.  THIS TYPEOUT ALSO OCCURS AFTER EVERY ERROR
;TYPEOUT OR IF A TTY KEY 4 IS TYPED.
;
;6.4.21 TELETYPE TEST
;
;THE TELETYPE TEST PRINTS A SINGLE LINE CONSISTING OF THE
;CHARACTERS SPACE TO Z.  THE TEST WILL ALSO ECHO CHARACTERS
;AS TYPED.
;
```

```
;7.0 ODT EDITOR
;7.1     REQUESTING THE ODT EDITOR
;        TO ENTER THE ODT TYPE A CONTROL O ON
;        THE TTI. THIS CAN BE DONE AT ANY POINT IN THE
;        PROGRAM.
;7.2     RESPONSE
;        ON ENTERING THE ODT A CARRIGE RETURN, LINE FEED
;        AND AN @ IS TYPED ON THE TTO.
;
;7.3 CONVENTIONS AND SYMBOLS IN COMMAND LINES
; ======================================
;
; CR     PRESSING THE RETURN KEY IS REPRESENTED BY CR .
;
; LF     PRESSING THE LINE FEED KEY IS REPRESENTED BY LF .
;
;?       PRESSING AN ILLEGAL KEY CAUSES THE ODT TO RESPONSE WITH
;        A ?.
; ^      PRESSING TH EUP-ARROW KEY IS REPRESENTED BY ^ .
;
;@       ODT IS READY AND AT YOUR SERVICE.
;
;
;7.4 COMMAND STRUCTURE
;  ==================
;
;        AN ODT COMMAND HAS THE GENERAL FORMAT:
;
;                [ARGUMENT][COMMAND]
;
;        ARGUMENT MAY BE ONE OF THE FOLLOWING:
;
;        ADR      AN OCTAL ADDRESS OR AN EXPRESSION OF THE FORM:
;                 X+X+X....
;                 WHERE EACH X IS AN OCTAL INTEGER, SEPARATED
;                  FROM THE FOLLOWING X BY EITHER +(PLUS)
;                 OR -(MINUS). LEADING ZEROS NEED NOT BE TYPED.
;
;        N        AN OCTAL INTEGER.
;
;        A COMMAND IS A SINGLE TELETYPE CHARACTER
;
;        CHARACTERS USED TO OPEN/CLOSE LOCATIONS INCLUDE:
;        "/"  "CR"  "LF"  "^"
;
;        CHARACTERS USED TO ENTER/EXIT ODT INCLUDE:
;        "^O"(CTRL O)  "R"  "P"
;
;        CHARACTERS USED TO MODIFY CURRENT ARGUMENTS ARE:
;        "RUBOUT"  "+" "-" AND THE INTEGERS 0 TO 7
;
;        THE CHARACTER "=" ALLOWS THE CURRENT ARGUMENT TO BE
;        EXAMINED WITHOUT OPENING OR CLOSING THE CURRENT LOC.
;        CHARACTERS USED TO MANIPULATE THE ECLIPSE MAP INCLUDE:
;        "M"  "A"  "B"  "U"  "T" "E" "L"
;
```



```
;
;7.5 COMMANDS TO OPEN A LOCATION
;  ======================
;
;
;        THE MEMORY LOCATION TO BE OPENED IS TYPEDOUT.
;ADR/    OPEN THE LOCATION AND PRINT ITS CONTENTS
;./      OPEN THE LOCATION CURRENTLY POINTED BY THE POINTER
;         AND PRINIT ITS CONTENTS.
;.+ADR/  ADD ADR TO THE POINTER, OPEN THE LOCATION AND
;        PRINT ITS CONTENTS.
;.-ADR/  SUBTRACT ADR FROM THE POINTER, OPEN THE LOCATION AND
;        PRINT ADDR CONTENTS.
; CR     CLOSE THE OPEN LOCATION WITH OR WITHOUT
;        MODIFICATION OF ITS CONTENTS.
; LF     CLOSE THE OPEN LOCATION WITH OR WITHOUT
;        MODIFICATION OF ITS CONTENTS AND OPEN THE
;        SUCCEDING LOCATION.
; /      CLOSE THE OPEN LOCATION WITHOUT MODIFYING
;        ITS CONTENTS AND OPEN THE CELL POINTED
;        BY ITS CONTENTS
;+ADR/   CLOSE THE OPEN LOCATION WITHOUT MODIFYING
;        ITS CONTENTS AND OPEN THE LOCATION POINTED
;        BY ITS CONTENTS+ADR
;-ADR/   CLOSE THE OPEN LOCATION WITHOUT MODIFYING ITS
;        CONTENTS AND OPEN THE LOCATION POINTED BY
;        ITS CONTENTS-ADR.
; ^      CLOSE THE CURRENT LOCATION AND OPEN ".-1"
;
```

```
!0045  .MAIN
                ;
                ;7.6 OTHER COMMANDS
                ;  --------------
                ;
                ;RUBOUT  THE RUBOUT KEY IS USED TO DELETE ERRONEOUSLY TYPED
                ;        DIGITS EACH TIME THE RUBOUT KEY IS PRESSED, THE RIGHT
                ;        MOST DIGIT IS DELETED AND ECHOED ON THE TERMINAL.
                ;        IF THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A CELL
                ;        THEN IT ALLOWS THE MODIFICATION OF THE CONTENTS AS IF
                ;        THEY WERE TYPED JUST BEFORE THE KEY WAS PRESSED.
                ;
                ; P      RESTART THE EXECUTION OF THE PROGRAM AT THE LOCATION
                ;        POINTED BY SAVED AC3 .
                ;
                ; ADRR   START EXECUTING THE USERS PROGRAM AT LOCATION ADR AFTER
                ;        AN IO RESET.
                ;
                ; K      KILL THE STRING TYPED SO FAR. ODT RESPONDS WITH A ? AND
                ;        THE OPEN LOCATION IS CLOSED WITHOUT MODIFICATION.
                ; =      PRINT THE CURRENT ARGUMENT  (I.E.  TYPING ".=" WILL
                ;        PRINT THE ADRS OF THE LAST OPENED LOCATION)
                ;
                ; M      SETS SWITCH SO THAT ALL MEMORY ACCESSES ARE
                ;        MAPPED USING THE LAST USER'S SETUP UNTIL A "U"
                ;        "E","T","A" OR "B"  IS INPUTTED.
                ;
                ; A      SETS SWITCH SO THAT ALL FURTHER MEMORY ACCESSES
                ;        ARE MATPED USER-A UNTIL A "B","U","E" OR "P" IS
                ;        ENTERED.
                ;
                ; B      SETS SWITCH SO THAT ALL FURTHER MEMORY ACCESSES
                ;        ARE MAPPED USING USER-B UNTIL A "A","T","U",OR "E"
                ;        IS ENTERED.
                ;
                ; T      PRINTS THE CURRENT CONTENTS OF ALL MAP ENTRY TABLES
                ;        AND CLEARS USE MAP SWITCH SO FURTHER MEMORY ACCESSES
                ;        ARE UNMAPPED.
                ;
                ; NE     PRINTS THE MAP ENTRY CORRESPONDING TO THE VALUE OF
                ;        N TYPED, WHERE N IS THE LOGICAL PAGE ADDRESS.
                ;        IF NO VALUE FOR N IS ENTERED THEN THE NEXT
                ;        LOGICAL MAP ENTRY IS TYPED.
                ;        THE USE MAP SWITCH IS CLEARED SO THAT ALL FURTHER
                ;        MEMORY ACCESSES ARE MAPPED.
                ;
                ; NL     SETS MAP LAST BLOCK TO VALUE ENTERED .
                ;        OCTAL VALUES CAN RANGE FROM 0 TO 377.
                ;        COMMANDS "A,B" CAN BE USED TO MODIFY FORMAT IF CALLED
                ;        BEFORE "L" COMMAND.
                ;        USE MAP SWITCH IS CLEARED BY THIS COMMAND.
                ;
                ; U      CLEARS USE MAP SWITCH TO STOP MAPPING OF MEMORY ACCESSES.
```

```
!0046  .MAIN
                ;
                ;7.7 MODIFICATION OF A LOCATION
                ;  ----------------------
                ;
                ;        ONCE A LOCATION HAS BEEN OPENED ITS CONTENTS CAN BE
                ;        MODIFIED IN ONE OF THE FOLLOWING WAYS:
                ;
                ;    1) TYPE THE OCTAL NUMBER OR A STRING OF NUMBERS SEPERATED
                ;       BY + OR -, FOLLOWED BY CR , OR LF . IN THIS CASE THE SUM
                ;       OF THE TOTAL NUMBERS TYPED-IN WILL BE DEPOSITED. LEADING
                ;       ZEROS NEED NOT BE TYPED.
                ;
                ;     2)        TYPE + OR - FOLLOWED BY A NUMBER OR A STRING OF NUMBERS
                ;       SEPERATED BY + OR -, FOLLOWED BY CR , OR LF . IN THIS
                ;       CASE SUM OF THE TOTAL NUMBERS TYPED IN WILL BE ADDED TO
                ;       OR SUBTRACTED FROM THE PREVIOUS CONTENTS OF THE LOCATION.
                ;       LEADING ZEROS NEED NOT BE TYPED.
                ;
                ;     3)        ADDRESS ITSELF OR AN OCTAL NUMBER RELATIVE TO THE
                ;       ADDRESS OF THE LOCATION CAN BE DEPOSITED IN A MEMORY
                ;       LOCATION BY TYPING A . OR .+ADR FOLLOWED BY A CR , OR LF .
                ;     4)        A RUBOUT COMMAND GIVEN RIGHT AFTER OPENING A
                ;       LOCATION ALLOWS THE MODIFICATION OF ITS CONTENTS
                ;       AS IF THEY WERE TYPEDIN JUST BEFORE THE COMMAND
                ;       WAS ISSUED.
                ;
```

**00025 TOTAL ERRORS, 00014 FIRST PASS ERRORS

ADRTS	000000	2/07	11/28	32/01				
ARITH	000000	2/09	11/14	31/01				
CATES	000000	2/22	20/01	41/19				
CBRDS	000000	2/06	10/22	29/23				
COMER	000000	2/13	14/17	34/01				
CXTES	000001	2/28						
DCUTS	000000	2/15	3/10	3/14	13/22	35/01		
EATS	000000	2/10	12/01	33/01				
EISTS	000000	2/08	11/01	30/01				
ERCCF	000001	2/05	4/43					
ERCOR	000000	2/04	25/01	29/04				
FPUTS	000000	2/11	12/13	33/09				
IOTST	000001	2/30	21/11	43/01				
LEFTS	000000	2/12	12/26	33/18				
LPTTS	000000	2/23	20/35	42/01				
MTTES	000000	2/21	19/01	41/19				
MVDSK	000000	2/19	6/29	7/25	17/01	40/01		
MXDSK	000001	2/27						
MXTES	000001	2/29						
NVDSK	000000	2/18	6/27	7/23	15/01	37/01		
N.DSK	000001	2/25						
PATCH	177777	3/14	6/27	6/28	6/29	6/30	7/23	7/24
		7/25	7/26					
PGDSK	000000	2/17	6/28	7/24	16/01	38/01		
PITTS	000000	2/31	21/01	42/16				
PZDSK	000000	2/20	6/30	7/26	18/01	41/14		
P.DSK	000001	2/24						
SCMTS	000000	2/14	13/01	35/13				
SHORT	000000	2/03						
SZDSK	000001	2/26						
WCSTS	000000	2/16	14/01	36/01				
ZLOAD	000001	2/33						