DataGeneral

TECHNICAL STATEMENT

TEXT LISTING

068-000272-05

PROGRAM

ECLIPSE MULTI-PROGRAMMING
RELIABILITY (SHORT)

TEXT TAPE

097-000272-05

ABSTRACT

THE ECLIPSE MULTI-PROGRAMMING RELIABILITY TEST (SHORT) 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.

```
 0001  .MAIN    AOS ASSEMBLER REV 02.02         09:47:59 09/28/78
01
02
03
04
05
06
07              ;*************************************************************
08              ;
09              ; NAME: EMORTS.TX                          PART NUMBER: 097-000272
10              ;
11              ; DESCRIPTION: ECLIPSE MULTI-PROGRAMMING RELIABILITY
12              ;                SHORT VERSION
13              ;
14              ; REVISION HISTORY:
15              ;
16              ;       REV.            DATE
17              ;
18              ;       04              09/01/77
19              ;       05              10/06/78
20              ;
21              ; COPYRIGHT (C) DATA GENERAL CORPORATION, 1975, 1976, 1977, 1978
22              ; ALL RIGHTS RESERVED.
23              ;*************************************************************
```

```
!0002  .MAIN
01              PATCH=0
02              ;FILE FOR EMORTV2 CPU TESTS ONLY
03       000001 SHORT=1
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       000001 PGDSK=1
18       000001 NVDSK=1
19       000001 MVDSK=1
20       000001 PZDSK=1
21       000001 MTTES=1
22       000001 CATES=1
23       000001 LPTTS=1
24       000001 P.DSK=1
25       000001 N.DSK=1
26       000001 MXDSK=1
27       000001 CXTES=1
28       000001 SZDSK=1
29       000001 MXTES=1
30       000000 IOTST=0
31       000000 PITTS=0
32       000000 .DUSR .MAPD=0
33       000000 ZLOAD=0
```

```
10003  .MAIN
01                ;       IF THE DCU 50 TEST IS TO BE RUN AT A DEVICE CODE
02                ;       OTHER TAN 64,ONE LOCATION HAS TO BE CHANGED SUCH
03                ;       IT CONTAINS THE DCU 50 DEVICE CODE.SEE DCU-50 TEST
04                ;       LOCATION DCUDV:
EEU       000000 PATCH=DCUDV
06                ;
07                ;3.       OPTIONAL STARTING ADDRESSES
08                ;3.1      200   AUTO-SIZE AND GO START
09                ;3.2      202   MANUAL SELECT/DELETE TESTS START
10                ;3.3      204   RUN UP TO FIRST 32K UNMAPPED START
11                ;3.4      206   RESTART LAST TEST SELECTIONS
12                ;3.5      210   START IN THE ODT
13                ;
14                ;       NOTE:   THIS PROGRAM CONTAINS A OCTAL DEBUGGER
15                ;       TO ASSIST IN GATHERING INFORMATION AT
16                ;       TIME OF ERROR AND PATCHING THE PROGRAM.
17                ;
18                ;         IF STARTED FROM DTOS AND THIS PROGRAM
19                ;       VERSION IS EMORT L OR EMORT P THEN IT
20                ;       WILL AUTOMATICALLY START AT ADDRESS
21                ;       210 TO ALLOW SELECTION OF TESTS/PATCHING
22                ;       AND OR OPERATOR SETUP. TYPE P TO SELECT
23                ;       A 202 START, OR TYPE 204R OR 200R FOR
24                ;       OTHER STARTS.
25                ;
```

```
10004  .MAIN
01                ;3.6    KEY ENTERED SWITCH OPTIONS(SWREG)
02                ;
03                ;               THE UTILIZATION OF KEYENTERED OPTIONS
04                ;               HAS BEEN IMPLEMENTED SUCH THAT THE
05                ;               CONSOLE "MONITOR" FUNCTION CAN
06                ;               BE FULLY UTILIZED WITHOUT INTERFERING
07                ;               WITH PROGRAM FLOW.
08                ;
09                ;  KEY 0        LOCKS THE SWPACKAGE INTO INPUT MODE
10                ;               ALLOWING SETUP OF THE CONTENTS OF "SWREG".
11                ;               TYPE A CARRAGE RETURN TO EXIT.
12                ;
13                ;       TYPING KEY'S 1 - 9,A - F SETS/RESETS SWREG BITS
14                ;       1 - 9,10 - 15 RESPECTIVELY. EACH KEY ENTRY
15                ;       COMPLEMENTS THE PREVIOUS STATE OF THE SWREG BIT.
16                ;
17                ; KEY   SWREG BIT       FUNCTION
18                ;
19                ; 1     1 = 1           DELETE TTO TYPEOUTS
20                ;
21                ; 2     2 = 1           DELETE MEM ALLOCATION TABLE
22                ;                       FROM TYPEOUTS
23                ;
24                ; 3     3 = 1           INCREASES THE CHANCE OF SELECTION
25                ;                       OF THE TTY AND LPT TEST.
26                ;
27                ; 4     4               TYPING A 4 WILL CAUSE THE ELAPSED RUN
28                ;                       TIME AND ACCUMULATED ERRORS
29                ;                       TO BE TYPED ON THE TTY.
30                ;                       (NOTE: A RTC MUST EXIST)
31                ;
32                ; 5     5 = 1           DIRECT ERROR AND RUNTIME TYPEOUTS
33                ;                       ALSO TO THE LINE PRINTER .
34                ;
35                ; 6     6 = 1           THE ERROR ROUTINE WILL PAUSE AFTER
36                ;                       EACH PHASE OF AN ERROR TYPEOUT.
37                ;                       TYPE A CR KEY ON DEVICE TTI TO PROCEED.
38                ;
39                ; 7     7               TYPING A 7 WILL CAUSE
40                ;                       INDIVIDUAL RUN STATISTICS OF
41                ;                       EACH TEST TO BE LISTED.
42                ;
43                ; F     15 = 1          DOESN'T RELEASE OR ALLOW REASIGNMENT
44                ;                       OF SCRATCH AREAS AFTER ERROR.
45                ;
46                ;       NOTE (C) = CONTROL KEY
47                ;
48                ;
49                ;       KEY M     PRINTS THE CURRENT CONTENNTS OF SWREG.
50                ;       KEY (C)D  RESTARTS THE PROGRAM AT LOC 202
51                ;                 AFTER RESETTING SWREG TO ALL 0'S
52                ;       KEY (C)R  RESTARTS THE PROGRAM AT LOC 202
53                ;                 WITHOUT DISTURBING SWREG.
54                ;
55                ; EXAMPLE:
56                ;       TO DELETE ERROR TYPEOUTS AND LOOP ON FAILING
57                ;       SCRATCH AREAS TYPE 0,1,F, AND CR.
```

```
10005  .MAIN
01                ;4.       OPERATING PROCEDURES
02                ;4.1      LOAD THE PROGRAM VIA THE BINARY LOADER
03                ;4.2      SET SWITCHES TO:
04                ;       200 FOR AUTO SIZE AND GO
05                ;       202 FOR MANUAL SELECT/DELETE
06                ;       204 TO IGNORE MMPU
07                ;       206 TO RESTART LAST SELECTED TESTS
08                ;NOTE:  THE TEST MUST HAVE BEEN STARTED
09                ;       AND RUN FOR A FEW SECONDS
10                ;       BEFORE ADRS 206 MAY BE UTILIZED
11                ;4.3      PRESS START
12                ;4.4      PROCESSOR WILL TYPE:
13                ;       ECLIPSE MULTI-PROG. REL. TEST XXX VERSIION
14                ;       TOTAL #1K'S=XXX(DECIMAL) MMPU,MMPU1 OR NO MAP
15                ;       PROGRAM RUN LIST
16                ;       PROG# DESCRIPTION
17                ;4.5      IF START WAS 200 (OR 206) THE LIST OF
18                ;       PROGRAMS TO BE RUN CONCURRENTLY WILL
19                ;       THEN BE LISTED AND THE TEST SYSTEM
20                ;       WILL AUTO START
21                ;4.6      IF START WAS 202 LINKER WILL
22                ;       PAUSE AT THE END OF EACH TEST
23                ;       DESCRIPTION AND WAIT FOR KEYBOARD
24                ;       INPUT.  TYPING IN A SPACE WILL
25                ;       ENABLE THAT TEST TO BE RUN.
26                ;       TYPING IN ANY OTHER CHARACTER WILL
27                ;       DELETE THAT TEST FROM BEING RUN
28                ;4.7      IF THE STARTING ADDRESS WAS 204 THE LINKER
29                ;       WILL SIZE MEMORY WITHOUT UTILIZING
30                ;       OR EVEN LOOKING FOR THE MAP OPTION
31                ;       AND THEN PROCEED AS IN STARTING AT ADRS. 202
32                ;       WITH THE MAP NONEXISTENT.
33                ;4.8    IF STARTED IN ODT (ADDRESS 210) AND "P"
34                ;       IS TYPED THE TEST WILL BE STARTED AS IF
35                ;       IT WAS STARTED AT 202.
36                ;4.9    OPTION SET UP
37                ;       IF A AUTOSTART ADDRESS WASN'T USED
38                ;       THEN THE PROGRAM WILL PAUSE AFTER PRINTING
39                ;       "ENTER OPTIONS,CR TO CONTINUE" AND ALLOW
40                ;       KEY ENTRY OF SWITCH REG. OPTIONS. TYPE
41                ;       A "CR" KEY TO START TESTS.
42                ;
```

```
10006  .MAIN
01                ;4.10   DUAL PORT DISK START UP PROCEDURE
02                ;
03                ; 4.10.1        PROGRAM LOAD EMORT(LONG OR PERIPHERAL)
04                ;               INTO EACH CPU.
05                ;
06                ; 4.10.2        INSTALL APPROPRIATE ODD/EVEN CYLINDER
07                ;               PATCHES INTO THE PROGRAM FOR EACH CPU.
08                ;               (SEE PARAG. 4.10.6)
09                ;
10                ; 4.10.3        START ONE CPU AT 202 ;WHEN QUERRED
11                ;               ABOUT TEST DELETIONS, START OTHER
12                ;               CPU AT 202.
13                ;
14                ; 4.10.4        WHEN QUERRED ABOUT TEST DELETIONS
15                ;               ON SECOND CPU ,ENTER ALL TEST
16                ;               SELECTIONS ON BOTH CPU'S.
17                ;
18                ; 4.10.5        ENTER ANY SWREG OPTIONS. CARRIAGE
19                ;               RETURN STARTS THE PROGRAM.
20                ;
21                ; 4.10.6        PATCH LIST
22                ;
23                ; 4.10.6.1      PRIMARY DEVICE CODE DISKS
24                ;
25                ;
26                ;LOCATION LABEL  TEST            ONE SYS OTHER SYS
27                PATCH=  ND.3L+6 ;NOVA DISK      101232  101233
28                PATCH=  PD.3L+6 ;6063/64 DISK   101232  101233
29                PATCH=  MH.3L+6 ;MOVING HEAD    101232  101233
30                PATCH=  Z8.3L+6 ;6060/61 DISK   101232  101233
31                ;
32                ; 4.10.6.2      SECONDARY DEVICE CODE DISKS
33                ;
34                ;LOCATION LABEL TEST            ONE SYS OTHER SYS
35                PATCH=  NX.3L+6 ;NOVA DISK      101232  101233
36                PATCH=  PX.3L+6 ;6063/64 DISK   101232  101233
37                PATCH=  MV.3L+6 ;MOVING HEAD    101232  101233
38                PATCH=  ZS.3L+6 ;6060/61 DISK   101232  101233
```

```
10007  .MAIN
01              ;
02              ;4.11   DATA LATE RETRIES FOR DISK TESTS
03              ;
04              ; DUE TO IMPLEMENTATION OF DISK CONFIGURATIONS WHICH WILL
05              ;RECEIVE DATA LATE STATUS ERRORS IN THEIR NORMAL COURSE
06              ;OF OPERATION, IF THEY ARE SECONDARY PRIORITY ON THE DATA
07              ;CHANNEL BUSS TO A DISK WHICH IS A FAST ACCESS TYPE, MULTI-
08              ;PROGRAMMING DISK TESTS NOW ARE CODED TO RETRY UP TO 16 TIMES
09              ;UPON DETECTION OF A DATA LATE ERROR. IF AFTER 16 RETRIES THE
10              ;ERROR STILL CONTINUES, THE ERROR WILL BE REPORTED. ALL
11              ;DATA LATE ERRORS WILL BE REPORTED AT SYSTEM STATUS REPORT
12              ;TIME, I.E. WHEN A "7" IS TYPED ON THE CONSOLE.
13              ;
14              ;4.11.1 PATCHES TO DISABLE DATA LATE RETRIES
15              ;
16              ; THE FOLLOWING PATCHES SHOULD BE INSTALLED IN THE DISK
17              ;TESTS WHICH LEGALLY SHOULDN'T RECEIVE ANY DATA LATE ERRORS.
18              ; THIS WILL DISABLE THE RETRY FEATURE FOR THAT TEST.
19              ;
20              ;4.11.1.1 PRIMARY DEVICE CODE DISK TESTS
21              ;
22              ;LOCATION LABEL TEST             WAS     CHANGE TO
23              PATCH=NDRTY     ;NOVA DISK      1000    401
24              PATCH=PD.SS-1   ;6063/64 DISK   403     401
25              PATCH=MHRWI+8.  ;MOVING HEAD    402     401
26              PATCH=ZBRTY     ;6060/61 DISK   402     401
27              ;
28              ;4.11.1.2 SECONDARY DEVICE CODE DISK TESTS
29              ;
30              ;LOCATION LABEL TEST             WAS     CHANGE TO
31              ;
32              PATCH=NXRTY     ;NOVA DISK      1000    401
33              PATCH=PX.SS-1   ;6063/64 DISK   403     401
34              PATCH=MVRWI+8.  ;MOVING HEAD    402     401
35              PATCH=ZSRTY     ;6060/61 DISK   402     401
```

```
10008  .MAIN
01              ;5.     ERROR DESCRIPTION
02              ;       MOST ERRORS DETECTED BY EITHER
03              ;       THE INDIVIDUAL TEST PROGRAMS OR
04              ;       BY THE DIAGNOSTIC LINKER WILL
05              ;       RESULT IN AN EXTENSIVE ERROR
06              ;       TYPEOUT.  SOME SMALL NUMBER OF
07              ;       HIGHLY IMPROBABLE ERRORS MAY RESULT
08              ;       IN A PROGRAM HALT IF THEY ARE
09              ;       OF A NATURE THAT THE LINKER CANNOT
10              ;       RECOVER FROM AND LOGICALLY PROCEED,
11              ;       (I.E. INTERRUPT STACK OVERFLOWS)
12              ;NOTE:  AFTER 50 ERRORS HAVE BEEN DETECTED, THE
13              ;       TEST WILL INDICATE SUCH VIA TYPEOUT, THEN
14              ;       WAIT FOR A CR KEY FROM DEVICE TTI IF TYPEOUTS
15              ;       AREN'T DELETED(SW1=1)
16              ;
17              ;5.1    ERROR FORMAT
18              ;       ERROR TYPEOUTS INCLUDE
19              ;5.1.1  PROGRAM # AND NAME AT TIME OF ERROR
20              ;5.1.2  THE CURRENT CONTENT OF THE USER STACK
21              ;       IN UP TO 2 SETS OF 5 VARIABLES.
22              ;       EACH SET OF VARIABLES (1 STATE BLOCK) INCLUDE
23              ;       AC0 TO AC3 AND THE PC WITH CARRY.
24              ;NOTE: THE 2 STATE BLOCKS TYPED ARE THE FIRST
25              ;       AND LAST 5 ENTRIES ON THE USER STACK.
26              ;       IF THERE ARE SAY SEVEN ENTRIES ON THE STACK
27              ;       THESE STATE BLOCKS WILL OVERLAP AND FOR THIS
28              ;       REASON THE START ADRS OF EACH STATE BLOCK
29              ;       IS INCLUDED IN THE TYPEOUT
30              ;5.1.3  LOGICAL SCRATCH AND DATA CHANNEL LIMITS
31              ;5.1.4  MEMORY ALLOCATION TABLE
32              ;       PHYSICAL 1K PAGE# +LOGICAL ADDRESS
33              ;       +REL LOG. ADDR(IF RELOCATED)
34              ;5.1.5  CONTINUATION INFORMATION IN GROUPS
35              ;       OF 3 MEMORY LOCATIONS PERTINENT TO
36              ;       THE INDIVIDUAL TEST THAT FAILED
37              ;
38              ;5.1.6  THE CPU TESTS THAT RELOCATE/REMAP WILL INCLUDE
39              ;IN THEIR ERROR TYPEOUTS:
40              ;ST.LA  START/ERROR (RES.)
41              ;XXXXXX YYYYYY  ZZZZZZ
42              ;
43              ;ST.LA  THE LOGICAL START OF THE RELOCATED TEST LOOP
44              ;XXXXXX (I.E. THE LAST LCALL SETUL)
45              ;
46              ;START  THIS NUMBER INDICATES WHERE THE RESIDENT COPY
47              ;YYYYYY OF THE TEST LOOP MAY BE FOUND IN THE LISTING
48              ;
49              ;ERROR  THIS NUMBER INDICATES WHERE IN THE RESIDENT
50              ;ZZZZZZ COPY OF THE LISTING THE ERROR CALL MAY BE FOUND
51              ;       (FOR SOME VALIDITY TRAP ERRORS THIS NUMBER
52              ;       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
```

```
!0011  .MAIN
01              ;5.3.2  EIS/MRI TEST
02              ;
03              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
04              ;       TIME OF ERROR DETECTION
05              ;
06              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
07              ;       EI.TK   SEE DISCUSSION OF TEST FOR THE
08              ;               SEQUENCE BEING EXECUTED
09              ;       EI.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
10              ;               REMAPPED FOR EXECUTION
11              ;       EI.LA   LOGICAL START OF TEST AFTER REMAP
12              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
13              ;
14              ;5.3.3  ARITHMETIC TEST
15              ;
16              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
17              ;       TIME OF ERROR DETECTION
18              ;
19              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
20              ;       AT.LC   STARTING ADDRESS OF ARITH IN SCRATCH
21              ;       AT.LO   LOW LIMIT OF SCRATCH AREA AFTER IT IS
22              ;               REMAPPED FOR EXECUTION
23              ;       AT.LA   AT.LC IN RELATION TO AT.LO
24              ;               (LOGICAL START OF ARITH AFTER REMAPPING)
25              ;       THE LAST THREE RANDOM NUMBERS GENERATED
26              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
27              ;
28              ;5.3.4  ADDRESS TEST
29              ;
30              ;       THE AC'S 0 1 AND 2 WILL
31              ;        BE TYPED AS THEY
32              ;       WERE AT THE TIME OF ERROR
33              ;       AC0 WILL CONTAIN THE ADDRESS THAT COMPARE WILL
34              ;       STOP AT (AC0<AC2=COMPARE IS TOP TO BOTTOM)
35              ;       AC1 WILL CONTAIN THE DATA FOUND TO BE INCORRECT
36              ;       AC2 WILL CONTAIN THE ADDRESS OR COMPLIMENT
37              ;               OF THE ADDRESS THAT FAILED
38              ;       AC3 WILL CONTAIN THE CURRENT RANDOM OFFSET
39              ;       IN ADDITION THE FOLLOWING LOC.'S WILL BE TYPED:
40              ;       A.TSK   TEST COUNTER
41              ;       0       ADRS. TO ADRS LOADED LOW TO HIGH
42              ;       2       COM. ADRS TO ADRS LOW TO HIGH
43              ;       4       ADRS. TO ADRS LOADED HIGH TO LOW
44              ;       6       COM. ADRS TO ADRS LOADED HIGH TO LOW
45              ;       10      ADRS TO ADRS LOADED LOW TO HIGH
46              ;       1,5, AND 11 COMPARE EACH ADRS. TO EQUAL ITSELF
47              ;       3 AND 7 COMPARE EACH ADRS. TO EQUAL ITS COM.
48              ;       AD.ST   STARTING ADDRESS OF TEST IN CORE
49              ;       AD.S3   ADRS. (AC3) OF ERROR CALL
```

```
!0012  .MAIN
01              ;5.3.5  EXTENDED ADDRESSING TEST TEST
02              ;
03              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
04              ;       TIME OF ERROR DETECTION
05              ;
06              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
07              ;       LA.LC   START ADRS. OF TEST
08              ;               FOR RELOCATION
09              ;       LA.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
10              ;               REMAPPED FOR EXECUTION
11              ;       LA.LA   LOGICAL START OF TEST AFTER REMAP
12              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
13              ;5.3.6  FLT PT TEST
14              ;
15              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
16              ;       TIME OF ERROR DETECTION
17              ;
18              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
19              ;       FP.LC   START ADRS. OF FLT. PT. TEST
20              ;               FOR RELOCATION
21              ;       FP.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
22              ;               REMAPPED FOR EXECUTION
23              ;       FP.LA   LOGICAL START OF TEST AFTER REMAP
24              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
25              ;
26              ;5.3.7  LEF/ERROR TEST
27              ;
28              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
29              ;       TIME OF ERROR DETECTION
30              ;
31              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
32              ;       LF.LC   START ADRS. OF TEST
33              ;               FOR RELOCATION
34              ;       LF.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
35              ;               REMAPPED FOR EXECUTION
36              ;       LF.LA   LOGICAL START OF TEST AFTER REMAP
37              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
```

```
10013  .MAIN
01              ;5.3.8  SC MEMORY TEST
02              ;
03              ; THIS IS AN ISZ/DSZ TEST FOR SC-MEMORIES.
04              ;
05              ; THE AC'S AT ERROR WILL INDICATE:
06              ; ACTUAL-EXPECTED-LOGICAL ADDRESS
07              ;
08              ; IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
09              ;       MM.TK   ERROR NUMBER:
10              ;       0       PATTERN STORING ERROR(SHD BE -1)
11              ;       1       LOCATION NOT -1 BEFORE DOING ISZ
12              ;       2       ISZ DIDN'T SKIP
13              ;       3       LOCATION NOT EQUAL TO 0 AFTER ISZ
14              ;       4       DSZ SKIP ERROR
15              ;       5       DSZ TEST-LOCATION NOT -1 AFTER DSZ
16              ;       6       SAME AS 1, EXCEPT TESTING IN REV DIRECTION
17              ;       7       SAME AS 2, EXCEPT   "      "  "     "
18              ;       10      SAME AS 3, EXCEPT   "      "      "      "
19              ;       MM.SE   INSTRUCTION ADDRESS FOLLOWING ERROR CALL
20              ;       LOCATION  ADDRESS OF FAILING LOCATION(LOGICAL)
21              ;
22              ;5.3.9  DCU TEST
23              ;
24              ;       THIS IS AN ARITHMETIC TEST PERFORMED BY THE
25              ;       DCU-50 USING THE DATA CHANNEL.
26              ;
27              ;       THE AC'S AT THE TIME OF ERROR DETECTION WILL
28              ;       BE TYPED .
29              ;       IN ADDITION THE FOLLOWING DATA IS TYPED:
30              ;        RANDOM DATA AC0,AC1,AC2
31              ;        DCLPK  LOGICAL START OF LOOP
32              ;        DCLER  LOGICAL ERROR ADDR
33              ;        DC.LA  LOGICAL START OF TEST
34              ;        DC.LP  LISTING START OF TEST
35              ;        ERROR  LISTING ADDR OF ERROR
```

```
10014  .MAIN
01              ;5.3.10 WCS TEST
02              ;
03              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE TIME
04              ;       OF ERROR DETECTION .
05              ;
06              ;
07              ;       IN ADDITION THE FOLLOWING LOCTIONS ARE TYPED:
08              ;
09              ;       WS.LC   STARTING ADDRESS OF WCS TEST IN SCRATCH
10              ;       WS.LO   LOW LIMIT OF SCRATCH AREA AFTER IT IS
11              ;               REMAPPED FOR EXECUTION.
12              ;       WS.LA   WS.LC IN RELATION TO WS.LO
13              ;               (LOGICAL START OF WCS TEST AFTER REMAPPING)
14              ;       THE LAST THREE NUMBERS GENERATED
15              ;       (SEE DISCUSSION OF ST.LA,ETC. AT PARAGRAPH 5.1.6)
16              ;
17              ;5.3.11 COMMERCIAL INSTRUCTION TEST
18              ;
19              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
20              ;       TIME OF ERROR DETECTION
21              ;
22              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
23              ;       CM.LC   START ADRS OF TEST(RELOCATED)
24              ;       CM.LO   LOW LIMIT OF SCRATCH AREA AFTER IT
25              ;               WAS REMAPPED FOR EXECUTION
26              ;       CM.LA   LOGICAL START OF TEST AFTER REMAP
27              ;       (SEE DISCUSSION OF ST.LA,ETC. AT PARA. 5.1.6)
```

```
10015  .MAIN
01              ;5.3.19 PROGRAMMABLE INTERVAL TIMER
02              ;
03              ;IF THE RTC AND PIT ARE RUN TOGETHER, THE PIT WILL MONITOR
04              ;AND UPDATE THE RTC IF ANY DIFFERENCE EXISTS.  IF A DIFFERENCE
05              ;OF MORE THAN 3 SECONDS IS FOUND IN ANY ONE MINUTE INTERVAL,
06              ;THE FOLLOWING ERROR MESSAGE WILL BE PRINTED:
07              ;       RTC AND PIT OUT OF SYNC XX SECONDS
08              ;IF EITHER THE RTC OR THE PIT ARE DISABLED, THEN THE PIT TEST
09              ;WILL GENERATE NO ERROR MESSAGES
10              ;
11              ;5.3.20 I/O TESTER TEST
12              ;
13              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE TIME
14              ; OF ERROR DETECTION.
15              ;
16              ; IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
17              ;
18              ; IO.OC COMMAND BEING PERFORMED
19              ; IO.BK # OF BLK'S READ/WRITTEN
20              ; B#ERR BLK # IN ERROR
21              ; ERRDST LOGICAL ADDR. OF LOC CONTAINING BAD DATA
22              ; IOCST BEGINNING DCH ADDRESS
23              ;
24              ; NOTE: IF THE OPERATION BEING PERFORMED WAS A
25              ; WRITE( BIT 1 OF IO.OC = 0) AND IF
26              ; A) AC2 IS WITHIN SCRLO/HI LIMITS
27              ;  THEN ERROR IS DUE TO CHECK OF OUTPUT
28              ;  BUFFER AREA AFTER A WRITE COMMAND.
29              ; B) AC2 IS NOT WITHIN SCRLO/HI LIMITS
30              ;  THEN ERROR IS AN XOR RESULT ERROR.
31              ;
```

```
10016  .MAIN
01              ;
02              ;5.4    SPECIAL CASE ERROR TYPEOUTS
03              ;
04              ;5.4.1  POWER FAIL INTERRUPT
05              ;       UPON DETECTION OF A POWER FAIL INTERRUPT
06              ;       THE LOGICAL ADRS. OF THE P.C. AT INTERRUPT
07              ;       WILL BE SAVED.
08              ;       IF AUTO-RESTART  IS ENABLED OR THE POWER
09              ;       FAIL WAS ONLY MOMENTARY, THE TEST WILL RE-
10              ;       START AS IN A START AT 206 AFTER TYPING
11              ;POWER FAIL @XXXXXX (WHERE XXXXXX IS THE PC AT INTR.)
12              ;
13              ;5.4.2  DATA CHANNEL PROTECT ERROR
14              ;       IF BIT 0 OF THE MMPU DIC STATUS
15              ;       REGISTER IS EVER DETECTED TO BE A 1,
16              ;       THE TEST WILL ATTEMPT TO ACCUMULATE
17              ;       AN INDICATION OF THE FREQUENCY OF THE
18              ;       ERRORS.
19              ;       PROG #  INDICATES THE TEST BEING EXECU-
20              ;               TED WHEN THE DCH ERRORS WHERE
21              ;               DETECTED.(MAY BE ANY TEST
22              ;               AS DCH'S OCCUR IN THE BACKGROUND)
23              ;       U ERRS  INDICATES THE NUMBER OF ERRORS
24              ;               THAT COULD BE DETERMINED TO OCCUR
25              ;               WITH THE MMPU IN "USER MODE"
26              ;       S ERRS  INDICATES THE NUMBER OF ERRORS THAT
27              ;               COULD BE DETERMINED TO OCCUR
28              ;               WITH THE MMPU IN "SUPERVISOR MODE"
29              ;NOTE: IT IS PROBABLY SAFE TO ASSUME THAT THE
30              ;       PERIPHERAL THAT IS THE RECIPIENT OF THESE
31              ;       DCH ERRORS WILL INTERJECT A FEW DATA
32              ;       ERROR TYPEOUTS OF IT'S OWN.
```

```
10017  .MAIN
01              ;
02              ;5.4.3  ILLEGAL SUPERVISOR CALL
03              ;
04              ;       THERE ARE SEVERAL CASES OF ILLEGAL SUPER-CALL'S
05              ;       1.) IF THE LOCATION INDICATED IS 000001
06              ;         THEN THE PROGRAM DETECTED THAT FOR SOME REASON
07              ;         LOCATION 000000 WAS EXECUTED. THE ILLEGAL
08              ;         SUPERCALL WAS FORCED BY THE ERROR HANDLER
09              ;         TO INDICATE WHICH TEST WAS EXECUTING AT THE
10              ;         TIME OF ERROR AND ITS ASSOCIATED INFORMATION.
11              ;       2.) IF OTHER THAN LOCATION 000001 THEN
12              ;         THE SUPERCALL HANDLER WAS ENTERED AND UPON
13              ;         CHECKING THE LOCATION FOLLOWING THE SUPER-
14              ;         CALL THERE WASN'T A VALID "JSR @XXXX".
15              ;         THIS COULD HAPPEN IN SEVERAL WAYS:
16              ;        A.) THE PROGRAM COULD HAVE JUMPED INTO
17              ;          A DATA AREA WHICH CONTAINED A WORD
18              ;          RECOGNIZED AS A SUPERCALL. IN THIS
19              ;          CASE AC3 OF THE SECOND ENTRY OF THE
20              ;          STACK USUALLY ISN'T 177510(SYC 3,3).
21              ;        B.) THE JSR @XXXX WORD WAS OVER WRITTEN
22              ;          BY ANOTHER TEST.
23              ;        C.) THE 177510(SYC 3,3) OR THE JSR @XXXX
24              ;          WAS IN A BAD MEMORY AREA CAUSING THE
25              ;          WORDS TO DROP OR ADD BITS.
26              ;        D.) THE JSR @XXXX WAS IN A DIFFERENT MAP
27              ;          PAGE WHOSE CONTENNTS ISN'T CORRECT.
28              ;        E.) SOME OTHER REASON ??
29              ;
30              ;  IN ANY CASE THE FOLLOWING INFORMATION IS TYPED:
31              ;       THE STACK SHOULD CONTAIN TWO BLOCKS.
32              ;       THE FIRST IS PUSHED BY THE ILLEGAL CALL.
33              ;        (IF SYC 3,3 THEN HEADER  IS CORRECT)
34              ;       THE SECOND IS BUILT BY THE ILLEGAL CALL
35              ;       HANDLER AND INCLUDES THE FOLLOWING INFO.
36              ;       AC0     ILLEGAL CALL IN LOC. XXXXX
37              ;       AC3     CONTENTS OF LOC. XXXXX-1
38              ;               (SHOULD = SYC 3,3 OR 177510)
```

```
10018  .MAIN
01              ;
02              ;5.4.4  ERROR TRAP
03              ;       AN I/O,WRITE,DEFER OR VALIDITY TRAP
04              ;       OCCURED THAT WAS NOT FORCED BY ANY TEST
05              ;       AND DID NOT = A READS 0 INSTRUCTION
06              ;       THE STATE BLOCK TYPED IS THAT PUSHED AS A
07              ;       RESULT OF THE TRAP
08              ;       IN ADDITION THE FOLLOWING ARE TYPED:
09              ;DIA MAP        CONTENTS OF THE MAP STATUS "A"
10              ;               (USER-LEF-WP,ETC.)
11              ;DOB            THE LAST BLOCK ALWAYS MAPPED FEATURE
12              ;               IS UTILIZED TO RETRIEVE THE INSTR.
13              ;               THAT TRAPPED.
14              ;5.4.4.1 OLD MAP DESCRIPTION
15              ;DIC            MAP STATUS "C" AS FOLLOWS
16              ;       BIT 0   DCH PROTECT
17              ;       BIT 1   SINGLE REFERENCE TRAP
18              ;       BIT 2   WRITE PROTECT TRAP
19              ;       BIT 3   VALIDITY PROTECT TRAP
20              ;       BIT 4   DEFER VIOLATION
21              ;       BIT 5   I/O PROTECT
22              ;       BIT 6   USER A/B
23              ;       BIT 7   P.C. UNDEFINED
24              ;       BITS 9-15 PHYSICAL BLK #
25              ;                 (377=VALIDTY PROTECT)
26              ;
27              ; 5.4.4.2 NEW MAP DESCRIPTION
28              ; DIA           MAP STATUS AS FOLLOWS
29              ;       BIT 1   EXT. FAULT
30              ;       BIT 2   I/O PROTECT TRAP
31              ;       BIT 3   WRITE PROTECT TRAP
32              ;       BIT 4   DEFER PROTECT TRAP
33              ;       BIT 5   SINGLE REF. TRAP
34              ;       BIT 6-8 FORMAT BITS
35              ;       BIT 9   LEF MODE
36              ;       BIT 10  I/O PROTECT ENABLE
37              ;       BIT 11  WRITE PROTECT ENABLE
38              ;       BIT 12  DEFER PROTECT ENABLE
39              ;       BIT 13  A/B USER SEL.
40              ;       BIT 14  DCH ENABLE
41              ;       BIT 15  USER MODE
42              ;
43              ; DIC   MAP STATUS "C" AS FOLLOWS
44              ;       BIT 0   WRITE PROTECT BIT
45              ;       BIT 1-3 FORMAT BITS
46              ;       BIT 5   EXT PAGE
47              ;       BITS 6-15 PHYSICAL BLK #
48              ;                 (1777=VALIDTY PROTECT)
49              ;
50              ;5.4.5  INTERRUPT WAIT ELAPSED
51              ;       THE PERIPHERAL DEVICE ASSOCIATED WITH THE
52              ;       PROG. NUMBER TYPED HAS NOT RESPONDED WITH
53              ;       A PROGRAM INTERRUPT FOR AN EXTENDED
54              ;       PERIOD OF TIME. THE 2ND NUMBER TYPED
55              ;       SHOULD POINT AT THE INTERRUPT HANDLER
56              ;       FOR THE DEVICE THAT FAILED
```

```
10019  .MAIN
01                ;
02                ;5.4.6  ERROR CORRECTION INTERRUPTS
03                ;       IF ERCC IS ENABLED THE TYPE OUTS THAT
04                ;       OCCUR WILL MORE LOGICALLY BE DUE TO
05                ;       ERROR CORRECTION INTERRUPTS THAN
06                ;       TO DATA FAILURES DETECTED BY
07                ;       ANY OF THE GOOD/BAD COMPARES
08                ;       AGAIN, THE INITIAL TYPEOUT WILL CONSIST
09                ;       OF THE FOUR AC'S:
10                ;       AC0= DATA WORD AFTER CORRECTION
11                ;
12                ;    IF OLD MAP(MMPU):
13                ;
14                ;       AC1= DIB FROM ERCC OPTION
15                ;               BITS 0 TO 4 ARE CORRECTION CODE
16                ;               BIT 15 INDICATES LOWER 64K IF=0
17                ;                      UPPER 64K IF=1
18                ;       AC2= PHYSICAL ADDRESS OF THE ERROR
19                ;               (WITHIN 64K INDICATED)
20                ;
21                ;    IF NEW MAP(MMPU1):
22                ;
23                ;       AC1= DIB FROM ERCC
24                ;         BITS 0-4 ARE CORRECTION CODE
25                ;         BITS 13-15 ARE THE COMPLEMENT
26                ;               OF HIGH ORDER PHYS ADDR BITS
27                ;       AC2= DIA FROM ERCC
28                ;         BITS 0-3 ARE COMPLEMENT OF PHYS
29                ;               ADDR BITS XPA2,PA1-3
30                ;         BITS 12-15 ARE THE COMPLEMENT
31                ;               OF PHYS ADDR BITS PA12-15
32                ;
33                ;       AC3= ADDRESS(PC) OF THE INTERRUPT
34                ;               (REFER TO LISTING)
35                ;       THE CORRECTION CODE IS RIGHT JUSTIFIED
36                ;       AND TYPED
37                ;       THE TOTAL NUMBER OF ERROR CORRECTION
38                ;       INTERRUPTS THAT OCCURED DURING THE LAST
39                ;       TEST LOOP ARE TYPED
```

```
10020  .MAIN
01                ;
02                ;FOR CONVENIENCE THE FOLLOWING TABLE
03                ;IS INCLUDED TO HELP DETERMINE WHICH BIT
04                ;IS FAILING IN AN ERROR CORRECTION MEMORY
05                ;COR.CODE       BIT
06                ;  0            NO ERROR
07                ;  1            CHECK BIT 4
08                ;  2            CHECK BIT 3
09                ;  3            DATA BIT 0
10                ;  4            CHECK BIT 2
11                ;  5            DATA BIT 1
12                ;  6            MULTIPLE BIT
13                ;  7            DATA BIT 3
14                ;  10           CHECK BIT 1
15                ;  11           DATA BIT 4
16                ;  12           ALL 21 BITS WERE=1
17                ;  13           DATA BIT 6
18                ;  14           DATA BIT 7
19                ;  15           DATA BIT 8
20                ;  16           DATA BIT 9
21                ;  17           MULT. BITS FAILED
22                ;  20           CHECK BIT 0
23                ;  21           DATA BIT 11
24                ;  22           DATA BIT 12
25                ;  23           DATA BIT 13
26                ;  24           DATA BIT 14
27                ;  25           ALL 21 BITS READ AS 0'S
28                ;  26           DATA BIT 2
29                ;  27           MULTIPLE BIT'S
30                ;  30           DATA BIT 10
31                ;  31           MULTIPLE BITS
32                ;  32           DATA BIT 5
33                ;  33           MULTIPLE BITS
34                ;  34           DATA BIT 15
35                ;  35           MULTIPLE BITS
36                ;  36           SAME
37                ;  37           SAME
38                ;
39                ;THE DATA TYPED BY THE ERROR CORRECTION
40                ;TYPEOUT IS AFTER CORRECTION HAS OCCURED.
41                ;THEREFORE, FOR THE BIT TO HAVE FAILED,
42                ;IT WOULD HAVE BEEN READ AS THE COMPLIMENT
43                ;OF HOW IT APPEARS IN THE TYPEOUT
44                ;
45                ;NOTE THAT ALL SINGLE BIT ERROR CODES
46                ;    INDICATE A CORRECTION BIT FAILED
```

```
!0017  .MAIN
01                ;
02                ;5.4.3  ILLEGAL SUPERVISOR CALL
03                ;
04                ;       THERE ARE SEVERAL CASES OF ILLEGAL SUPER-CALL'S
05                ;       1.) IF THE LOCATION INDICATED IS 000001
06                ;         THEN THE PROGRAM DETECTED THAT FOR SOME REASON
07                ;         LOCATION 000000 WAS EXECUTED. THE ILLEGAL
08                ;         SUPERCALL WAS FORCED BY THE ERROR HANDLER
09                ;         TO INDICATE WHICH TEST WAS EXECUTING AT THE
10                ;         TIME OF ERROR AND ITS ASSOCIATED INFORMATION.
11                ;       2.) IF OTHER THAN LOCATION 000001 THEN
12                ;         THE SUPERCALL HANDLER WAS ENTERED AND UPON
13                ;         CHECKING THE LOCATION FOLLOWING THE SUPER-
14                ;         CALL THERE WASN'T A VALID "JSR @XXXX".
15                ;         THIS COULD HAPPEN IN SEVERAL WAYS:
16                ;        A.) THE PROGRAM COULD HAVE JUMPED INTO
17                ;          A DATA AREA WHICH CONTAINED A WORD
18                ;          RECOGNIZED AS A SUPERCALL. IN THIS
19                ;          CASE AC3 OF THE SECOND ENTRY OF THE
20                ;          STACK USUALLY ISN'T 177510(SYC 3,3).
21                ;        B.) THE JSR @XXXX WORD WAS OVER WRITTEN
22                ;          BY ANOTHER TEST.
23                ;        C.) THE 177510(SYC 3,3) OR THE JSR @XXXX
24                ;          WAS IN A BAD MEMORY AREA CAUSING THE
25                ;          WORDS TO DROP OR ADD BITS.
26                ;        D.) THE JSR @XXXX WAS IN A DIFFERENT MAP
27                ;          PAGE WHOSE CONTENNTS ISN'T CORRECT.
28                ;        E.) SOME OTHER REASON ??
29                ;
30                ;  IN ANY CASE THE FOLLOWING INFORMATION IS TYPED:
31                ;       THE STACK SHOULD CONTAIN TWO BLOCKS.
32                ;       THE FIRST IS PUSHED BY THE ILLEGAL CALL.
33                ;        (IF SYC 3,3 THEN HEADER  IS CORRECT)
34                ;       THE SECOND IS BUILT BY THE ILLEGAL CALL
35                ;       HANDLER AND INCLUDES THE FOLLOWING INFO.
36                ;       AC0     ILLEGAL CALL IN LOC. XXXXX
37                ;       AC3     CONTENTS OF LOC. XXXXX+1
38                ;               (SHOULD = SYC 3,3 OR 177510)
```

```
!0018  .MAIN
01                ;
02                ;5.4.4  ERROR TRAP
03                ;       AN I/O,WRITE,DEFER OR VALIDITY TRAP
04                ;       OCCURED THAT WAS NOT FORCED BY ANY TEST
05                ;       AND DID NOT = A READS 0 INSTRUCTION
06                ;       THE STATE BLOCK TYPED IS THAT PUSHED AS A
07                ;       RESULT OF THE TRAP
08                ;       IN ADDITION THE FOLLOWING ARE TYPED:
09                ;DIA MAP        CONTENTS OF THE MAP STATUS "A"
10                ;               (USER+LEF+WP,ETC.)
11                ;DOB            THE LAST BLOCK ALWAYS MAPPED FEATURE
12                ;               IS UTILIZED TO RETRIEVE THE INSTR.
13                ;               THAT TRAPPED.
14                ;5.4.4.1 OLD MAP DESCRIPTION
15                ;DIC            MAP STATUS "C" AS FOLLOWS
16                ;       BIT 0   DCH PROTECT
17                ;       BIT 1   SINGLE REFERENCE TRAP
18                ;       BIT 2   WRITE PROTECT TRAP
19                ;       BIT 3   VALIDITY PROTECT TRAP
20                ;       BIT 4   DEFER VIOLATION
21                ;       BIT 5   I/O PROTECT
22                ;       BIT 6   USER A/B
23                ;       BIT 7   P.C. UNDEFINED
24                ;       BITS 9-15 PHYSICAL BLK #
25                ;                 (377=VALIDTY PROTECT)
26                ;
27                ; 5.4.4.2 NEW MAP DESCRIPTION
28                ; DIA            MAP STATUS AS FOLLOWS
29                ;       BIT 1   EXT. FAULT
30                ;       BIT 2   I/O PROTECT TRAP
31                ;       BIT 3   WRITE PROTECT TRAP
32                ;       BIT 4   DEFER PROTECT TRAP
33                ;       BIT 5   SINGLE REF. TRAP
34                ;       BIT 6-8 FORMAT BITS
35                ;       BIT 9   LEF MODE
36                ;       BIT 10  I/O PROTECT ENABLE
37                ;       BIT 11  WRITE PROTECT ENABLE
38                ;       BIT 12  DEFER PROTECT ENABLE
39                ;       BIT 13  A/B USER SEL.
40                ;       BIT 14  DCH ENABLE
41                ;       BIT 15  USER MODE
42                ;
43                ; DIC   MAP STATUS "C" AS FOLLOWS
44                ;       BIT 0   WRITE PROTECT BIT
45                ;       BIT 1-3 FORMAT BITS
46                ;       BIT 5   EXT PAGE
47                ;       BITS 6-15 PHYSICAL BLK #
48                ;                 (1777=VALIDTY PROTECT)
49                ;
50                ;5.4.5  INTERRUPT WAIT ELAPSED
51                ;       THE PERIPHERAL DEVICE ASSOCIATED WITH THE
52                ;       PROG. NUMBER TYPED HAS NOT RESPONDED WITH
53                ;       A PROGRAM INTERRUPT FOR AN EXTENDED
54                ;       PERIOD OF TIME. THE 2ND NUMBER TYPED
55                ;       SHOULD POINT AT THE INTERRUPT HANDLER
56                ;       FOR THE DEVICE THAT FAILED
```

```
!0019  .MAIN
01                ;
02                ;5.4.6  ERROR CORRECTION INTERRUPTS
03                ;       IF ERCC IS ENABLED THE TYPE OUTS THAT
04                ;       OCCUR WILL MORE LOGICALLY BE DUE TO
05                ;       ERROR CORRECTION INTERRUPTS THAN
06                ;       TO DATA FAILURES DETECTED BY
07                ;       ANY OF THE GOOD/BAD COMPARES
08                ;       AGAIN, THE INITIAL TYPEOUT WILL CONSIST
09                ;       OF THE FOUR AC'S:
10                ;       AC0= DATA WORD AFTER CORRECTION
11                ;
12                ;    IF OLD MAP(MMPU):
13                ;
14                ;       AC1= DIB FROM ERCC OPTION
15                ;               BITS 0 TO 4 ARE CORRECTION CODE
16                ;               BIT 15 INDICATES LOWER 64K IF=0
17                ;                       UPPER 64K IF=1
18                ;       AC2= PHYSICAL ADDRESS OF THE ERROR
19                ;               (WITHIN 64K INDICATED)
20                ;
21                ;    IF NEW MAP(MMPU1):
22                ;
23                ;       AC1= DIB FROM ERCC
24                ;         BITS 0-4 ARE CORRECTION CODE
25                ;         BITS 13-15 ARE THE COMPLEMENT
26                ;               OF HIGH ORDER PHYS ADDR BITS
27                ;       AC2= DIA FROM ERCC
28                ;         BITS 0-3 ARE COMPLEMENT OF PHYS
29                ;               ADDR BITS XPA2,PA1-3
30                ;         BITS 12-15 ARE THE COMPLEMENT
31                ;               OF PHYS ADDR BITS PA12-15
32                ;
33                ;       AC3= ADDRESS(PC) OF THE INTERRUPT
34                ;               (REFER TO LISTING)
35                ;       THE CORRECTION CODE IS RIGHT JUSTIFIED
36                ;       AND TYPED
37                ;       THE TOTAL NUMBER OF ERROR CORRECTION
38                ;       INTERRUPTS THAT OCCURED DURING THE LAST
39                ;       TEST LOOP ARE TYPED
```

```
!0020  .MAIN
01                ;
02                ;FOR CONVENIENCE THE FOLLOWING TABLE
03                ;IS INCLUDED TO HELP DETERMINE WHICH BIT
04                ;IS FAILING IN AN ERROR CORRECTION MEMORY
05                ;COR.CODE       BIT
06                ;  0            NO ERROR
07                ;  1            CHECK BIT 4
08                ;  2            CHECK BIT 3
09                ;  3            DATA BIT 0
10                ;  4            CHECK BIT 2
11                ;  5            DATA BIT 1
12                ;  6            MULTIPLE BIT
13                ;  7            DATA BIT 3
14                ;  10           CHECK BIT 1
15                ;  11           DATA BIT 4
16                ;  12           ALL 21 BITS WERE=1
17                ;  13           DATA BIT 6
18                ;  14           DATA BIT 7
19                ;  15           DATA BIT 8
20                ;  16           DATA BIT 9
21                ;  17           MULT. BITS FAILED
22                ;  20           CHECK BIT 0
23                ;  21           DATA BIT 11
24                ;  22           DATA BIT 12
25                ;  23           DATA BIT 13
26                ;  24           DATA BIT 14
27                ;  25           ALL 21 BITS READ AS 0'S
28                ;  26           DATA BIT 2
29                ;  27           MULTIPLE BIT'S
30                ;  30           DATA BIT 10
31                ;  31           MULTIPLE BITS
32                ;  32           DATA BIT 5
33                ;  33           MULTIPLE BITS
34                ;  34           DATA BIT 15
35                ;  35           MULTIPLE BITS
36                ;  36           SAME
37                ;  37           SAME
38                ;
39                ;THE DATA TYPED BY THE ERROR CORRECTION
40                ;TYPEOUT IS AFTER CORRECTION HAS OCCURED.
41                ;THEREFORE, FOR THE BIT TO HAVE FAILED,
42                ;IT WOULD HAVE BEEN READ AS THE COMPLIMENT
43                ;OF HOW IT APPEARS IN THE TYPEOUT
44                ;
45                ;NOTE THAT ALL SINGLE BIT ERROR CODES
46                ;    INDICATE A CORRECTION BIT FAILED
```

```
10021  .MAIN
01              ;
02              ;6.2    PROGRAM INITIALIZE
03              ;THE DIAGNOSTIC LINKER INITIALIZES ITSELF
04              ;AND INDIVIDUAL TESTS IN THE FOLLOWING
05              ;SEQUENCE:
06              ;   1. SYSTEM IS RESET, MAP OPTION IS
07              ;      DETERMINED TO EXIST OR NOT EXIST
08              ;      AND SWITCHES ARE SET UP
09              ;      ACCORDINGLY
10              ;   2. ANY OTHER NECESSARY CONSTANTS
11              ;        ARE INITIALIZED
12              ;      (MEM ALLOCATION TABLES)
13              ;   3. INTERRUPT VECTOR TABLES ARE SET UP TO
14              ;      PROCESS UNEXPECTED DEVICE INTERRUPTS
15              ;   4. MEMORY IS SIZED IN 1K INCREMENTS
16              ;      FROM 0 TO 128K AND BUILDS AN 8 WORD
17              ;      BIT MAP OF EXISTING CONTIGUOUS
18              ;      MEMORY(256K=16 WORDS FOR MMPU1).
19              ;
20              ;   5. THE EXIST MAP IS MOVED TO THE
21              ;      AVAILABLE MAP AND EACH BIT
22              ;      CORRESPONDING TO 1K OF UITILIZED
23              ;      MEMORY IS REMOVED FROM THE MAP
24              ;      SO THAT IT WILL NOT BE ASSIGNED
25              ;      AS A SCRATCH AREA TO ANY TEST.
26              ;      (INCLUDES PROGRAM STORAGE, MEMORY ALLOC.
27              ;      TABLES, INTERRUPT MASKS AND STACK AREA AND
28              ;      THE LAST 1K OF MEMORY TO PRESERVE THE
29              ;      BINARY LOADER)
30              ;   6. EACH TEST IS ENTERED IN SEQUENCE AT ITS
31              ;      INIT. ENTRY POINT. OPTION TESTS DETERMINE
32              ;      IF THE DEVICE THEY ARE ASSOC. WITH EXISTS
33              ;      OR NOT AND PASS INTERRUPT SERVICE PARAM'S
34              ;      TO THE LINKER.
35              ;        (DEV#, MASK AND INTERRUPT SERVICE
36              ;      ADDRESS)
37              ;   7. LINKER THEN TYPES THE SYSTEM SIZE
38              ;      INFORMATION ALONG WITH THE PROGRAM
39              ;      RUN LIST AND WILL ALLOW THE OPERATOR
40              ;      TO SELECT OR DELETE SPECIFIC TESTS
41              ;       AND ENTER KEY OPTIONS
42              ;      IF START WAS 00202 OR 204.
43              ;   8. AFTER STARTING, THOSE TESTS THAT HAVE
44              ;   "SIZED" THEIR SUBSYSTEM FOR SPECIFIC
45              ;   PARAMETERS TYPE AN INDICATION OF THE PARAMETERS
46              ;   THEY DETERMINED TO EXIST.(SEE THE
47              ;   INDIVIDUAL DISK TEST DESCRIPTIONS.)
```

```
10022  .MAIN
01              ;6.3    PROGRAM RUN
02              ;       ONCE THE LINKER HAS COMPLETED ALL
03              ;       INITIALIZATION THE FOLLOWING SERIES
04              ;       OF OPERATIONS IS LOOPED THROUGH
05              ;
06              ;          1. LINKER RANDOMLY SELECTS ONE OF
07              ;             THE INDIVIDUAL TESTS UNTIL IT
08              ;             FINDS ONE THAT IS NOT WAITING
09              ;             FOR INTERRUPT (WAIT IS BIT 0 OF
10              ;             THE THIRD WORD IN TEST=1) AND THAT
11              ;             THE NEXT RANDOM NUMBER FALLS WITHIN
12              ;             ITS ENTER LIMITS
13              ;
14              ;          2. IF THE MAP OPTION EXISTS, ALL LOGICAL PAGES
15              ;             EXCEPT PAGE 0 ARE ACCESS PROTECTED WITH
16              ;             THE PHYSICAL AREA OF THE SELECTED TEST
17              ;             MAPPED TO ITSELF AND ANY ASSIGNED
18              ;             SCRATCH AREA MAPPED TO START AT 1K
19              ;             ABOVE THE TEST, MEMORY LOCATIONS SCRLO
20              ;             AND SCRHI (SCRATCH LOW AND HIGH) ARE
21              ;             SET TO INDICATE THE LIMITS OF
22              ;             THE SCRATCH AREA AVAILABLE TO THE TEST.
23              ;
24              ;          3. DATA CHANNEL LIMITS (DCHLO AND DCHHI)
25              ;             ARE CALCULATED AND ENTERED
26              ;
27              ;          4. THE SELECTED TEST IS ENTERED AT
28              ;             ITS SPECIFIED EXECUTE ENTRY POINT
```

```
!0023  .MAIN
01
02              ;6.4    INDIVIDUAL TEST DESCRIPTIONS
03              ;
04              ;6.4.1  ERROR CORRECTION
05              ;
06              ;  THE ERROR CORRECTION OPTION IN MULTIPROGRAMMING
07              ; NORMALLY ENABLES ERROR CORRECTION IN MODE 3 (CORRECT
08              ;AND INTERRUPT).  IF AN ERROR CORRECTION INTERRUPT OCCURS THIS
09              ;TEST SAVES ALL INFORMATION AVAILABLE PERTAINING TO THE FIRST
10              ;ERROR FOR LATER TYPEOUT. BETWEEN THE TIME OF THE ERROR CORRECTION
11              ;INTERRUPT AND COMPLETION OF THE TYPEOUT ERROR CORRECTION IS
12              ;RE-ENABLED IN "MODE 3". HOWEVER, UNTIL THE TYPEOUT IS COMPLETED,
13              ;ERROR CORRECTION INTERRUPTS ARE ONLY COUNTED
14              ;NOTE:  THE TYPEOUT DOES NOT OCCUR IMMEDIATELLY AT THE TIME
15              ;       OF INTERRUPT, IT IS INITIATED AFTER
16              ;       COMPLETION OF THE CURRENT PASS OF THE TEST THAT
17              ;       WAS RUNNING WHEN THE INTERRUPT OCCURRED. ALSO,
18              ;       IF THE ADDRESS OF THE ERROR WAS OUTSIDE OF THE
19              ;       SCRATCH AND RESIDENT AREAS OF THE TEST THAT WAS
20              ;       RUNNING, ONE SHOULD SUSPECT THAT THE ERROR
21              ;       COULD HAVE OCCURRED DURING A DATA CHANNEL.
22              ;
23              ;6.4.2  CHECKERBOARD RAN
24              ;
25              ;THIS MEMORY CHECKER BOARD TEST IS A SUBSET OF OTHER MEMORY
26              ;CHECKERBOARDS.  A COMPLETE TEST OF AN AVAILABLE SCRATCH
27              ;AREA IS COMPRISED OF THE FOLLOWING SEQUENCE:
28              ;
29              ;CB.TK=0        ;REQUEST 1 TO 32K OF SCRATCH, RANDOMLY RE-
30              ;               ;LOCATE THE EXECUTE PORTION OF CHECKERBOARD
31              ;               ;INTO SCRATCH AND GENERATE THE CHECKERBOARD
32              ;               ;PATTERN
33              ;
34              ;CB.TK=1        ;DISTURB PASS-COMPLIMENT A SINGLE BIT IN EACH
35              ;               ;OF THE FIRST 16 WORDS OF SCRATCH, SHUFFLE THESE
36              ;               ;WORDS 16 TIMES SUCH THAT THEY END UP IN THEIR
37              ;               ;ORIGINAL POSITION, RE-COMPLIMENT THE SINGLE
38              ;               ;BIT IN EACH WORD-PROCEED WITH EACH GROUP OF
39              ;               ;16 WORDS UNTIL ALL MEMORY HAS BEEN EXERCISED.
40              ;
41              ;CB.TK=2        ;CHECK PASS-COMPARE EACH WORD IN SCRATCH WITH
42              ;               ;THE PATTERN EXPECTED
43              ;
44              ;CB.TK=3        ;FAST CHECKSUM MEMORY TO ENSURE THAT ALL DATA
45              ;               ;IS INTACT (RETURNS TO CHECK PASS IF CHECK-
46              ;               ;SUM DOES NOT AGREE.)
```

```
!0024  .MAIN
01              ;6.4.3  EIS/MRI TEST
02              ;
03              ;THE MULTIPROGRAMMING EXTENDED INSTRUCTION SET - MEMORY
04              ;REFERENCE INSTRUCTION TEST IS ESSENTIALLY ANOTHER MEMORY
05              ;CHECKERBOARD EXERCISER.  THIS TEST, HOWEVER,INCLUDES 19
06              ;BIT PATTERNS FOR ERROR CORRECTION MEMORY'S AND UTILIZES
07              ;BYTE, BIT AND BLM INSTRUCTIONS TO EXERCISE THE MEM-
08              ;ORY.  A COMPLETE PASS OF THE EIS/MRI TEST INCLUDES ALL
09              ;OF THE FOLLOWING:
10              ;
11              ;EI.TK=0        ;ASSIGN SCRATCH (1 TO 32K), RANDOMLY RE-
12              ;               ;LOCATE EXECUTABLE CODE INTO THE SCRATCH AREA,
13              ;               ;GENERATE CHECKERBOARD PATTERN.(1 OF 4 RANDOM
14              ;               ;SELECTED IF ERROR CORRECTION)
15              ;
16              ;EI.TK=1        ;VERIFY THAT THE SCRATCH AREA CONTAINS THE
17              ;               ;CORRECT CHECKERBOARD PATTERN
18              ;
19              ;EI.TK=2        ;RANDOMLY SELECT GROUPS OF 16 WORDS COMLIMENT
20              ;               ;A SINGLE BIT IN EACH WORD, SHUFFLE 16 WORDS
21              ;               ;16 TIMES, RECOMPLIMENT THE SINGLE BIT IN
22              ;               ;EACH WORD.
23              ;
24              ;EI.TK=3        ;(SEE EI.TK=1) COMPARE
25              ;
26              ;EI.TK=4        ;SELECT RANDOM X DRIVERS, COMPLIMENT A
27              ;               ;SINGLE BIT ON EACH OF 16 DRIVERS (EVERY 64TH WORD)
28              ;               ;WORD) BLM ALL OF THE PATTERN AREA BACK TO ITSELF,
29              ;               ;RECOMPLIMENT BITS
30              ;EI.TK=5                ;(SEE EI.TK=1) COMPARE
31              ;EI.TK=6-7-10   ;BLM THE PATTERN AREA TO ITSELF
32              ;EI.TK=11       ;(SEE EI.TK=1) COMPARE
33              ;EI.TK=12       ;BYTE TEST - LDB - COMPLIMENT
34              ;               ;STB LDB - RE-COMPLIMENT - STB EACH
35              ;               ;SEQUENTIAL BYTE IN THE PATTERN AREA.
36              ;EI.TK=13       ;(COMPARE AGAIN)
37              ;EI.TK=14       ;GENERATE COMPLIMENT WORST CASE PATTERN
38              ;EI.TK=15       ;(SEE EI.TK=1) COMPARE
39              ;EI.TK=16       ;(SEE EI.TK=2) BIT TEST
40              ;EI.TK=17       ;(SEE EI.TK=3) COMPARE
41              ;EI.TK=20       ;(SEE EI.TK=4) DRIVERS
42              ;EI.TK=21       ;(SEE EI.TK=5) COMPARE
43              ;EI.TK=22,23,23 ;(SEE EI.TK=6,7,10) BLM'S
44              ;EI.TK=25       ;(SEE EI.TK=11) COMPARE
45              ;EI.TK=26       ;(SEE EI.TK=12) BYTE'S
46              ;EI.TK=27       ;(SEE EI.TK=13) COMPARE
```

```
10025  .MAIN
01                ;6.4.4  ARITHMETIC TEST
02                ;
03                ;THE MULTIPROGRAMMING RELIABILITY ARITHMETIC TEST WAS
04                ;DERIVED FROM THE STAND ALONE ARITHMETIC TEST.  THIS TEST
05                ;REQUIRES 2K OF SCRATCH FOR EXECUTION.  THE EXECUTE POR-
06                ;TION OF THE TEST IS RANDOMLY RELOCATED WITHIN AVAILABLE
07                ;SCRATCH.  IF THE SYSTEM IS MAPPED, (HAS AN MMPV) THE
08                ;SCRATCH AREA IS RANDOMLY REMAPPED TO SOME OTHER LOGICAL AD-
09                ;DRESS FOR EXECUTION.  AT THE END OF EACH EXECUTION PASS SCRATCH
10                ;AREA IS RANDOMLY RELEASED OR HELD.  IF HELD, THE NEXT TIME
11                ;THE TEST IS ENTERED, THE EXECUTABLE PORTION OF THE TEST WILL
12                ;AGAIN BE RANDOMLY RELOCATED WITHIN SCRATCH FOR EXECUTION.
13                ;
```

```
10026  .MAIN
01                ;6.4.5  ADDRESS TEST
02                ;
03                ;THE MULTIPROGRAMMING ADDRESS TEST GENERATES A VARIETY OF
04                ;4 ADDRESS AND COMPLIMENT ADDRESS PATTERNS IN AVAILABLE
05                ;SCRATCH.  THE TEST SEQ FOLLOWS:
06                ;
07                ;A.TSK=0        ;SELECT A SCRATCH AREA TO WORK IN, MOVE THE
08                ;               ;EXECUTE PORTION OF THE ADDRESS TEST TO
09                ;               ;EITHER THE HIGH OR LOW END OF SCRATCH
10                ;               ;GENERATE THE ADDRESS TO ADDRESS PAT-
11                ;               ;TERN IN THE UPWARD (LOW SCRATCH TO HIGH
12                ;               ;SCRATCH) DIRECTION, THE PATTERN IS GEN-
13                ;               ;ERATED IN TWO STEPS.  FIRST A RANDOM # OF
14                ;               ;WORDS (OFFSET A.AOS) IS FILLED UIA A BAM
15                ;               ;(ADRS(AC2))+1(AC0) TO ADRS(AC3) THE
16                ;               ;REST OF SCRATCH IS FILLED VIA A SECOND BAM
17                ;               ;(ADRS (AC2))+(OFFSET+1(AC0) TO ADRS AC3.
18                ;
19                ;A.TSK=1        ;VERIFY THAT EACH SCRATCH LOCATION CONTAINS
20                ;               ;ITS OWN LOGICAL ADDRESS.
21                ;
22                ;A.TSK=2        ;PATTERN GENERATION IS PERFORMED AS IN
23                ;               ;A.TSK=0 EXCEPT THAT THE 2 BAM'S FILL EACH
24                ;               ;SCRATCH LOCATION WITH THE 1'S COMPLIMENT OF
25                ;               ;ITS LOGICAL ADDRESS.
26                ;
27                ;A.TSK=3        ;VERIFY THAT EACH LOCATION CONTAINS THE 1'S
28                ;               ;COMPLIMENT OF ITS LOGICAL ADDRESS.
29                ;
30                ;A.TSK=4        ;EACH SCRATCH LOCATION IS AGAIN FILLED WITH
31                ;               ;ITS OWN LOGICAL ADDRESS.  HOWEVER, THE PATTERN IS
32                ;               ;GENERATED IN THE DOWNWARD DIRECTION (HIGH SCRATCH
33                ;               ;TO LOW SCRATCH) TWO STEPS ARE TAKEN.  A RANDOM
34                ;               ;# OF WORDS EQUAL TO OFFSET (A.AOS) ARE FILLED
35                ;               ;WITH THEIR LOGICAL ADRS.  THE REST OF SCRATCH IS
36                ;               ;THEN FILLED VIA AN ELDA AC2+OFFSET, OFFSET IS
37                ;               ;SUBTRACTED AND THE AN STH INTO THE LOWER
38                ;               ;LOCATION,
39                ;
40                ;A.TSK=5        ;(SEE A.TSK=1)
41                ;
42                ;A.TSK=6        ;GENERATION IS AS A.TSK=4 EXCEPT EACH LOCATION
43                ;               ;IS FILLED WITH THE 1'S COMPLIMENT OF ITS
44                ;               ;LOGICAL ADDRESS
45                ;
46                ;A.TSK=7                ;(SEE A.TSK=3)
47                ;
48                ;A.TSK=10       ;THE PATTERN GENERATION OF A.TSK=0 IS RERUN
49                ;
50                ;A.TSK=11       ;(SEE A.TSK=1)
```

```
!0027  .MAIN
01              ;6.4.6  EXTENDED ADDRESSING TEST
02              ;THE EXTENDED ADDRESSING TEST
03              ;VERIFIES THE CORRECT OPERATION OF THE DOUBLE LENGTH
04              ;INSTRUCTIONS,THE IMMEDIATE MODE DOUBLE LENGTH,
05              ;DISPA,CLM,MSP,HLV AND ELEF.
06              ;SCRATCH AREA IS TREATED AS INTHE EIS/MRI TEST
07              ;WITH TEST EXECUTION SIMILAR TO THE ARITH,FLT PT.
08              ;AND LEF/ERROR TESTS
09              ;6.4.7  FLOATING POINT TEST
10              ;
11              ;THE MULTIPROGRAMMING FLOATING POINT TEST IS SIMILAR IN OP-
12              ;ERATION TO THE ARITHMETIC TEST.  THE FPU TEST DATA BUFFER IS
13              ;LOCATED RANDOMLY WITHIN 32 WORDS IN THE CENTER OF THE EX-
14              ;ECUTE PORTION OF THE FLT PT TEST.  THE FLT. PT. NUMBERS PRO-
15              ;CCESSED BY THIS TEST ARE FIXED (NOT RANDOM) AND CAN BE FOUND
16              ;SPECIFIED ON THE LISTING FOR EACH TEST.
17              ;
18              ;6.4.8  LEF/ERROR TEST
19              ;
20              ;THE LEF MODE - ERROR TEST IN MULTIPROGRAMMING RELIABILITY IS
21              ;ONLY RUN IF AN MMPU EXISTS.  THE LEF PORTION OF THIS TEST VERIFIES
22              ;THAT THE LEF ENABLE ON THE MMPU FUNCTIONS IN ALL ADDRESSING
23              ;MODES.  THE ERROR PORTION OF THE TEST VERIFIES THAT THE WRITE, I/O
24              ;DEFER AND VALIDITY PROTECT FEATURES OF THE MMPU FUNCTION
25              ;CORRECTLY.  SCRATCH AND EXECUTION ARE TREATED AS IN ARITH-
26              ;METIC TEST.
27              ;
```

```
!0028  .MAIN
01              ;6.4.9  COMMERCIAL INSTRUCTION TEST
02              ;
03              ;UPON ENTERING FOR INITIALIZATION THIS TEST DOES A TRIAL
04              ;INSTRUCTION TO DETERMINE IF THE COMMERCIAL OPTION AND
05              ;THE FLOATING POINT OPTION ARE INSTALLED TOGETHER.
06              ;UPON ENTERING FOR EXECUTION THE TEST TRIES TO ACQUIRE
07              ;4K OF SCRATCH. IF OBTAINED THE TEST MODULE IS
08              ;MOVED UP INTO SCRATCH LEAVING THE AREA ABOVE AND
09              ;BELOW THE TEST AS DATA SCRATCH BUFFERS.
10              ;THE SIX INSTRUCTIONS TESTED ARE: ELDB,ESTB,CMV,CMP
11              ;CTR, AND CMT.
12              ; THE EXTENDED LOAD BYTE,STORE BYTE INSTRUCTIONS ARE
13              ;TESTED TOGETHER. A RANDOM BYTE ADDRESS IS GENERATED IN
14              ;EITHER THE HIGH OR THE LOW SCRATCH BUFFER. A RANDOM
15              ;NUMBER (16 BITS) IS THEN STORED INTO MEMORY LOCATION
16              ;CONTAINING THAT BYTE. BITS 8-15 OF THIS RANDOM NUMBER
17              ;IS AGAIN STORED INTO THAT BYTE ADDRESS BY THE ESTB INSTR.
18              ; THESE TWO ADJACENT BYTES OF THE SAME WORD ARE THEN LOADED
19              ;ONE AT A TIME BY THE ELDB INSTRUCTION BACK INTO THE ACCUM.
20              ;THE RANDOM NUMBER IS HENCE RECONSTRUCTED AND COMPARED WITH
21              ;THE ORIGINAL FOR ERROR. THIS EXERCISE CHECKS THAT THE ELDB,
22              ; ESTB INSTRUCTIONS DO NOT DISTURB THE ADJACENT BYTES.
23              ;ALL FOUR ADDRESSING MODES ARE TESTED SEPERATELY.
24              ;  THE CMV, CMP, CTR, AND CMT INSTRUCTIONS ALL DEAL
25              ;WITH STRINGS OF BYTES FROM A SOURCE TO A DESTINATION
26              ;FIELD. THE SIZES OF THE LOW AND HIGH BUFFER ARE COMPARED
27              ;AND THE SMALLER BUFFER IS SELECTED AS THE SOURCE
28              ;BUFFER WHILE THE LARGER IS THE DESTINATION. THE SOURCE
29              ;BUFFER IS FILLED WITH RANDOM DATA AT THE START OF THE
30              ;TEST IN ORDER TO MINIMIZE THE USE OF THE RANDOM DATA
31              ;GENERATOR. THE CMV AND CMP INSTRUCTIONS ARE EXERCISED TOGETHER.
32              ;THE CTR AND CMT INSTR. ARE EXERCISED SEPERATELY
33              ;AND THEIR TRANSLATION TABLE IS LOCATED RANDOMLY IN THE SOURCE
34              ;BUFFER. SINCE THE CTR INSTRUCTION REQUIRES A TRANSLATION TBL.
35              ;OF 128 WORDS, A SCHEME IS SETUP WHICH DIVIDES THE
36              ;LARGER BUFFER INTO TWO HALVES IF THE SMALLER BUFFER
37              ;SIZE IS LESS THAN 256 WORDS. IN THIS CASE, THE LOWER HALF IS
38              ;CHOSEN AS THE SOURCE AND THE UPPER HALF AS THE DESTINATION
39              ;BUFFER.
40              ; THE LDI,STI,LDIX,LSN AND FINT INSTRUCTIONS HANDLE THE
41              ;CONVERSION OF INTEGERS AND FLOATING POINT NUMBERS.
42              ;A RANDOM NUMBER IS FLOATED FROM MEMORY INTTO FPAC0,
43              ;THE EXPONENT OF WHICH IS THEN RANDOMIZED. FPAC0 IS
44              ;INTEGERIZED (BY FINT) AND STORED (BY STI) AS AN
45              ;INTEGER 16 BYTES LONG. THIS INTEGER IS THEN LOADED BY
46              ;LDI INTO FPAC1, AND COMPARED WITH FPAC0 FOR ERROR.
47              ;THE SAME INTEGER IS ALSO TESTED BY THE LSN INSTRUCTION
48              ;AND THE RETURNED CODE IN AC1 IS EXAMINED FOR ERROR.
49              ;A SIMILAR APPROACH IS USED IN TESTING THE LDIX ,STIX
50              ;INSTRUCTIONS. ALL EIGHT FORMATS OF INTEGERS(TYPES 0-7)
51              ;ARE TESTED SEPERATELY.
52              ; THE EDIT INSTRUCTION IS USED TO CONVERT INTERGERS
53              ;FROM ONE FORMAT TO THE OTHER. THREE TESTS ARE
54              ;WRITTEN IN WHICH ALL EDIT OP CODES ARE EXERCISED.
```

```
10029  .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 DFLETED.
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              ;
```

```
10030  .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              ;
```

```
!0031  .MAIN
01              ;
02              ;6.4.19 PROGRAMMABLE INTERVAL TIMER TEST
03              ;
04              ;THE PROGRAMMABLE INTERVAL TIMER IS SET FOR A 1 SECOND INTERVAL.
05              ;ASSUMING THE PIT TO BE MORE RELIABLE THAN THE REAL TIME CLOCK,
06              ;THE PIT TEST WILL UPDATE THE RTC EVERY MINUTE AND INDICATE AN
07              ;ERROR IF THE TWO DIFFER BY MORE THAN 5% (3 SEC/MIN).  IF THE
08              ;RTC DOES NOT EXIST, THE PIT ASSUMES THE REAL TIME CLOCK'S
09              ;RESPONSIBILITY OF REPORTING THE TIME AT 5, 15, AND 30 MINUTES
10              ;AND EVERY 30 MINUTES THEREAFTER. THE TIME WILL ALSO BE INDICATED
11              ;AFTER EVERY ERROR.
12              ;
13              ;6.4.20 REAL TIME CLOCK
14              ;
15              ;THE REAL TIME CLOCK IS RUN AT 1K HERTZ.  RUNTIME ALONG
16              ;WITH ACCUMULATED ERROR COUNT ARE PRINTED AT 5 MINUTES
17              ;15 MINUTES, 30 MINUTES AND EVERY 30  MINUTES OF RUNTIME
18              ;THEREAFTER.  THIS TYPEOUT ALSO OCCURS AFTER EVERY ERROR
19              ;TYPEOUT OR IF A TTY KEY 4 IS TYPED.
20              ;
21              ;6.4.21 TELETYPE TEST
22              ;
23              ;THE TELETYPE TEST PRINTS A SINGLE LINE CONSISTING OF THE
24              ;CHARACTERS SPACE TO Z.  THE TEST WILL ALSO ECHO CHARACTERS
25              ;AS TYPED.
26              ;
```

```
!0032  .MAIN
01              ;6.4.22 I/O TESTER TEST
02              ;
03              ;THIS TEST UTILIZES THE I/O TESTER TO PERFORM
04              ;DATA CHANNEL OPERATIONS. IF THE SYSTEM
05              ;HAS A MMPU1(NEW MAP) INSTALLED AND THE I/O TESTER
06              ;IS AT THE PROPER ECO LEVEL BOTH DCH MAPS A AND
07              ;B WILL BE USED IN MAPPED MODE.
08              ;
09              ;THE I/O TESTER IS ASSUMED TO BE ASSIGNED TO DEVICE
10              ;CODE 40.
11              ;READ OR WRITE OPERATIONS ARE SELECTED RANDOMLY.
12              ;2 TO 4 K OF SCRATCH AREA ARE ASSIGNED TO THIS TEST.
13              ;READ OPERATION:
14              ;THE I/O TESTER BUFFER IS LOADED WITH A RANDOM WORD
15              ;AND THE BLK SIZE IS SET UP FOR 64 WORDS. READ COMMAND IS
16              ;GIVEN AND REPEATED UNTIL ENOUGH BLKS TO FILL SCRATCH
17              ;ARE RECEIVED. THEN SCRATCH AREA IS CHECKED FOR
18              ;THE PROPER DATA WORD.
19              ;WRITE OPERATION:
20              ;64 RANDOM WORDS ARE GENERATED AND THEN MOVED INTO
21              ;SCRATCH AREA REPEATEDLY UNTIL AREA IS FILLED.
22              ;WRITE COMMAND IS GIVEN AFTER SETTING BLK SIZE
23              ;TO 64 WORDS. THE IO TESTER INTERRUPTS AFTER EACH BLK
24              ;AND THE DATA BUFFER REGISTER IS READ AND STORED FOR
25              ;LATER CHECK. THIS IS REPEATED UNTIL ALL BLKS HAVE BEEN
26              ;RECEIVED. NEXT THE SCRATCH AREA IS CHECKED TO INSURE
27              ;IT WASN'T DISTURBED AND THEN A SOFTWARE XOR OF THE
28              ;64 WORDS OF DATA IS FORMED AND EACH XOR RESULT WORD
29              ;IS COMPARED TO IT.
```

```
!0033  .MAIN
01              ;7.0 ODT EDITOR
02              ;7.1    REQUESTING THE ODT EDITOR
03              ;       TO ENTER THE ODT TYPE A CONTROL O ON
04              ;       THE TTI. THIS CAN BE DONE AT ANY POINT IN THE
05              ;       PROGRAM.
06              ;7.2    RESPONSE
07              ;       ON ENTERING THE ODT A CARRIGE RETURN, LINE FEED
08              ;       AND AN @ IS TYPED ON THE TTO.
09              ;
10              ;7.3 CONVENTIONS AND SYMBOLS IN COMMAND LINES
11              ; ---------------------------------------
12              ;
13              ; CR    PRESSING THE RETURN KEY IS REPRESENTED BY CR .
14              ;
15              ; LF    PRESSING THE LINE FEED KEY IS REPRESENTED BY LF .
16              ;
17              ;?      PRESSING AN ILLEGAL KEY CAUSES THE ODT TO RESPONSE WITH
18              ;       A ?.
19              ; ^     PRESSING TH EUP-ARROW KEY IS REPRESENTED BY ^ .
20              ;
21              ;@      ODT IS READY AND AT YOUR SERVICE.
22              ;
23              ;
24              ;7.4 COMMAND STRUCTURE
25              ;  -----------------
26              ;
27              ;       AN ODT COMMAND HAS THE GENERAL FORMAT:
28              ;
29              ;               [ARGUMENT] [COMMAND]
30              ;
31              ;       ARGUMENT MAY BE ONE OF THE FOLLOWING:
32              ;
33              ;       ADR     AN OCTAL ADDRESS OR AN EXPRESSION OF THE FORM:
34              ;               X+X+X....
35              ;               WHERE EACH X IS AN OCTAL INTEGER, SEPARATED
36              ;                FROM THE FOLLOWING X BY EITHER +(PLUS)
37              ;               OR -(MINUS). LEADING ZEROS NEED NOT BE TYPED.
38              ;
39              ;       N       AN OCTAL INTEGER.
40              ;
41              ;       A COMMAND IS A SINGLE TELETYPE CHARACTER
42              ;
43              ;       CHARACTERS USED TO OPEN/CLOSE LOCATIONS INCLUDE:
44              ;       "/"  "CR"  "LF"  "^"
45              ;
46              ;       CHARACTERS USED TO ENTER/EXIT ODT INCLUDE:
47              ;       "^O"(CTRL O)  "R"  "P"
48              ;
49              ;       CHARACTERS USED TO MODIFY CURRENT ARGUMENTS ARE:
50              ;       "RUBOUT"  "+" "-" AND THE INTEGERS 0 TO 7
51              ;
52              ;       THE CHARACTER "=" ALLOWS THE CURRENT ARGUMENT TO BE
53              ;       EXAMINED WITHOUT OPENING OR CLOSING THE CURRENT LOC.
54              ;       CHARACTERS USED TO MANIPULATE THE ECLIPSE MAP INCLUDE:
55              ;       "M"  "A"  "B"  "U"  "T" "E" "L"
56              ;
```

```
!0034  .MAIN
01              ;
02              ;7.5 COMMANDS TO OPEN A LOCATION
03              ;  ----------------------
04              ;
05              ;
06              ;       THE MEMORY LOCATION TO BE OPENED IS TYPEDOUT.
07              ;ADR/   OPEN THE LOCATION AND PRINT ITS CONTENTS
08              ;./     OPEN THE LOCATION CURRENTLY POINTED BY THE POINTER
09              ;        AND PRINIT ITS CONTENTS.
10              ;.+ADR/ ADD ADR TO THE POINTER, OPEN THE LOCATION AND
11              ;       PRINT ITS CONTENTS.
12              ;.-ADR/ SUBTRACT ADR FROM THE POINTER, OPEN THE LOCATION AND
13              ;       PRINT ADDR CONTENTS.
14              ; CR    CLOSE THE OPEN LOCATION WITH OR WITHOUT
15              ;       MODIFICATION OF ITS CONTENTS.
16              ; LF    CLOSE THE OPEN LOCATION WITH OR WITHOUT
17              ;       MODIFICATION OF ITS CONTENTS AND OPEN THE
18              ;       SUCCEDING LOCATION.
19              ; /     CLOSE THE OPEN LOCATION WITHOUT MODIFYING
20              ;       ITS CONTENTS AND OPEN THE CELL POINTED
21              ;       BY ITS CONTENTS
22              ;+ADR/  CLOSE THE OPEN LOCATION WITHOUT MODIFYING
23              ;       ITS CONTENTS AND OPEN THE LOCATION POINTED
24              ;       BY ITS CONTENTS+ADR
25              ;-ADR/  CLOSE THE OPEN LOCATION WITHOUT MODIFYING ITS
26              ;       CONTENTS AND OPEN THE LOCATION POINTED BY
27              ;       ITS CONTENTS-ADR.
28              ; ^     CLOSE THE CURRENT LOCATION AND OPEN ".-1"
29              ;
```

```
!0035  .MAIN
01                ;
02                ;7.6 OTHER COMMANDS
03                ;  --------------
04                ;
05                ;RUBOUT THE RUBOUT KEY IS USED TO DELETE ERRONEOUSLY TYPED
06                ;       DIGITS EACH TIME THE RUBOUT KEY IS PRESSED, THE RIGHT
07                ;       MOST DIGIT IS DELETED AND ECHOED ON THE TERMINAL.
08                ;       IF THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A CELL
09                ;       THEN IT ALLOWS THE MODIFICATION OF THE CONTENTS AS IF
10                ;       THEY WERE TYPED JUST BEFORE THE KEY WAS PRESSED.
11                ;
12                ; P     RESTART THE EXECUTION OF THE PROGRAM AT THE LOCATION
13                ;       POINTED BY SAVED AC3 .
14                ;
15                ; ADRR  START EXECUTING THE USERS PROGRAM AT LOCATION ADR AFTER
16                ;       AN IO RESET.
17                ;
18                ; K     KILL THE STRING TYPED SO FAR. ODT RESPONDS WITH A ? AND
19                ;       THE OPEN LOCATION IS CLOSED WITHOUT MODIFICATION.
20                ; =     PRINT THE CURRENT ARGUMENT  (I.E.  TYPING ".=" WILL
21                ;       PRINT THE ADRS OF THE LAST OPENED LOCATION)
22                ;
23                ; M     SETS SWITCH SO THAT ALL MEMORY ACCESSES ARE
24                ;       MAPPED USING THE LAST USER'S SETUP UNTIL A "U"
25                ;       "E","T","A" OR "B"  IS INPUTTED.
26                ;
27                ; A     SETS SWITCH SO THAT ALL FURTHER MEMORY ACCESSES
28                ;       ARE MATPED USER-A UNTIL A "B","U","E" OR "P" IS
29                ;       ENTERED.
30                ;
31                ; B     SETS SWITCH SO THAT ALL FURTHER MEMORY ACCESSES
32                ;       ARE MAPPED USING USER-B UNTIL A "A","T","U",OR "E"
33                ;       IS ENTERED.
34                ;
35                ; T     PRINTS THE CURRENT CONTENTS OF ALL MAP ENTRY TABLES
36                ;       AND CLEARS USE MAP SWITCH SO FURTHER MEMORY ACCESSES
37                ;       ARE UNMAPPED.
38                ;
39                ; NE    PRINTS THE MAP ENTRY CORRESPONDING TO THE VALUE OF
40                ;       N TYPED, WHERE N IS THE LOGICAL PAGE ADDRESS.
41                ;       IF NO VALUE FOR N IS ENTERED THEN THE NEXT
42                ;       LOGICAL MAP ENTRY IS TYPED.
43                ;       THE USE MAP SWITCH IS CLEARED SO THAT ALL FURTHER
44                ;       MEMORY ACCESSES ARE MAPPED.
45                ;
46                ; NL    SETS MAP LAST BLOCK TO VALUE ENTERED .
47                ;       OCTAL VALUES CAN RANGE FROM 0 TO 377.
48                ;       COMMANDS "A,B" CAN BE USED TO MODIFY FORMAT IF CALLED
49                ;       BEFORE "L" COMMAND.
50                ;       USE MAP SWITCH IS CLEARED BY THIS COMMAND.
51                ;
52                ; U     CLEARS USE MAP SWITCH TO STOP MAPPING OF MEMORY ACCESSES.
```

```
!0036  .MAIN
01                ;
02                ;7.7 MODIFICATION OF A LOCATION
03                ;  ----------------------
04                ;
05                ;       ONCE A LOCATION HAS BEEN OPENED ITS CONTENTS CAN BE
06                ;       MODIFIED IN ONE OF THE FOLLOWING WAYS:
07                ;
08                ;    1) TYPE THE OCTAL NUMBER OR A STRING OF NUMBERS SEPERATED
09                ;       BY + OR -, FOLLOWED BY CR , OR LF . IN THIS CASE THE SUM
10                ;       OF THE TOTAL NUMBERS TYPED-IN WILL BE DEPOSITED. LEADING
11                ;       ZEROS NEED NOT BE TYPED.
12                ;
13                ;     2)        TYPE + OR - FOLLOWED BY A NUMBER OR A STRING OF NUMBERS
14                ;       SEPERATED BY + OR -, FOLLOWED BY CR , OR LF . IN THIS
15                ;       CASE SUM OF THE TOTAL NUMBERS TYPED IN WILL BE ADDED TO
16                ;       OR SUBTRACTED FROM THE PREVIOUS CONTENTS OF THE LOCATION.
17                ;       LEADING ZEROS NEED NOT BE TYPED.
18                ;
19                ;     3)        ADDRESS ITSELF OR AN OCTAL NUMBER RELATIVE TO THE
20                ;       ADDRESS OF THE LOCATION CAN BE DEPOSITED IN A MEMORY
21                ;       LOCATION BY TYPING A . OR .+ADR FOLLOWED BY A CR , OR LF .
22                ;     4)        A RUBOUT COMMAND GIVEN RIGHT AFTER OPENING A
23                ;       LOCATION ALLOWS THE MODIFICATION OF ITS CONTENTS
24                ;       AS IF THEY WERE TYPEDIN JUST BEFORE THE COMMAND
25                ;       WAS ISSUED.
26                ;
```

10037 .MAIN

**00003 TOTAL ERRORS, 00002 FIRST PASS ERRORS

0038 .MAIN

ADRTS	000000	2/07	11/28	26/01		
ARITH	000000	2/09	11/14	25/01		
CATES	000001	2/22	15/01	31/01		
CBRDS	000000	2/06	10/22	23/23		
COMER	000000	2/13	14/17	28/01		
CXTES	000001	2/27				
DCUTS	000000	2/15	3/01	3/05	13/22	29/01
EATS	000000	2/10	12/01	27/01		
EISTS	000000	2/08	11/01	24/01		
ERCCF	000001	2/05	4/43			
ERCOR	000000	2/04	19/01	23/04		
FPUTS	000000	2/11	12/13	27/09		
IOTST	000000	2/30	15/11	32/01		
LEFTS	000000	2/12	12/26	27/18		
LPTTS	000001	2/23	15/01	31/01		
MTTES	000001	2/21	15/01	31/01		
MVDSK	000001	2/19	15/01	31/01		
MXDSK	000001	2/26				
MXTES	000001	2/29				
NVDSK	000001	2/18	15/01	31/01		
N.DSK	000001	2/25				
PATCH	000000'	3/05				
PGDSK	000001	2/17	15/01	31/01		
PITTS	000000	2/31	15/01	31/02		
PZDSK	000001	2/20	15/01	31/01		
P.DSK	000001	2/24				
SCMTS	000000	2/14	13/01	29/13		
SHORT	000001	2/03				
SZDSK	000001	2/28				
WCSTS	000000	2/16	14/01	30/01		
ZLOAD	000000	2/33				