DataGeneral

TECHNICAL STATEMENT

TEXT LISTING

068-000646-01

PROGRAM

ECL MULTI-PROGRAMMING RELI TST
(SHORT)

TEXT TAPE

097-000646-01

ABSTRACT

THE ECLMORT ECLIPSE MULTIPROGRAMMING RELIABILITY TEST CONSISTS OF A SERIES OF INDIVIDUAL PROCESSOR AND PERIPHERAL TESTS AND A SUPERVISOR PROGRAM, THE DIAGNOSTIC LINKER.

```
 0001  SEMRT    AOS ASSEMBLER REV 02.02         20:01:55 03/01/79
01
02
03
04
05
06              ;*********************************************************
07              ;
08              ; NAME: ECLMORTS.TX                        P.N. 097-000646
09              ;
10              ; DESCRIPTION: ECLMORT(ECLIPSE MULTI-PROGRAMMING RELIABILITY
11              ;                TEST,  SHORT VERSION)
12              ;
13              ; REVISION HISTORY
14              ;
15              ;       REV.            DATE
16              ;
17              ;       00              09/15/78
18              ;       01              03/01/79
19              ;
20              ;
21              ; COPYRIGHT (C) DATA GENERAL CORPORATION 1978,1979
22              ; ALL RIGHTS RESERVED.
23              ;*****************************************************************
```

```
10002  SEMRT
01              ;*********************************************************
02              ;       CONDITIONAL ASSEMBLY FLAGS
03              ; 0= INCLUDE TEST
04              ; 1= DON'T INCLUDE TEST
05              ;*********************************************************
06              ;FILE FOR ECLMORT-S(CPU TESTS ONLY)
07       000001 SHORT=1
08       000000 ERCOR=0          ;ERROR CORRECTION TEST
09       000001 ERCCF=1
10       000000 CBRDS=0          ;CHECKER BOARDS RANDOM
11       000000 ADRTS=0          ;ADDRESS TEST
12       000000 EISTS=0          ;EIS/MRI TEST
13       000000 ARITH=0          ;ARITHMETIC TEST
14       000000 EATS=0           ;EXTENDED ADDRESSING TEST
15       000000 FPUTS=0          ;FLOATING POINT TEST
16       000000 FPFTS=0          ;FLOATING POINT FUNCTIONS TEST
17       000000 LEFTS=0          ;LEF/MAP ERROR TEST
18       000000 COMER=0          ;COMMERCIAL TEST
19       000000 SCMTS=0          ;SC MEMORY TEST
20       000000 DCUTS=0          ;DCU-50/200 TEST
21       000000 APTS=0           ;ARRAY PROCESSOR TEST
22       000000 WCSTS=0          ;WCS TEST
23       000001 PGDSK=1          ;PRIMARY PAGING DISK(6063/64) TEST
24       000001 MVDSK=1          ;PRIMARY MOVING HEAD DISK TEST
25       000001 PZDSK=1          ;PRIMARY ZEBRA DISK(6060 SERIES) TEST
26       000001 MTTES=1          ;PRIMARY MAGNETIC TAPE TEST
27       000001 LPTTS=1          ;PRIMARY LINE PRINTER TEST
28       000001 P.DSK=1          ;SECONDARY PAGING DISK(6063/64) TEST
29       000001 SZDSK=1          ;SECONDARY ZEBRA(6060 SERIES)DISK TEST
30       000001 MXDSK=1          ;SECONDARY MOVING HEAD DISK TEST
31       000001 MXTES=1          ;SECONDARY MAG TAPE TEST
32       000000 IOTST=0          ;I/O TESTER TEST
33       000000 ZLOAD=0
```

```
!0003  SEMRT
01                ; 0.    REVISION HISTORY
02                ;
03                ;       REV. 0  INITIAL RELEASE FOR M600,S130 SUPPORT
04                ;       REV. 1  UPDATE TO SUPPORT:
05                ;               IOP ARRAY PROCESSOR
06                ;               MULTIPLE IOP SUPPORT
07                ;               PIT TEST REMOVED
08                ;               DCU-200 SUPPORT ADDED
09                ;
10                ;1.     ABSTRACT
11                ;       THE (ECLMORT) ECLIPSE MULTIPROGRAMMING RELIABILITY TEST
12                ;       CONSISTS OF A SERIES OF INDIVIDUAL PROCESSOR
13                ;       AND PERIPHERAL TESTS AND A
14                ;       SUPERVISOR PROGRAM. (THE DIAGNOSTIC LINKER)
15                ;
16                ;       THE DIAGNOSTIC LINKER IS A PROGRAM
17                ;       DESIGNED TO "LINK" THE VARIETY OF
18                ;       PROCESSOR AND PERIPHERAL TESTS IN
19                ;       SUCH A FASHION THAT THEY MAY BE
20                ;       RUN CONCURRENTLY.  THEREBY, TESTING
21                ;       THE INTERACTIVE CAPABILITIES OF
22                ;       THE PROCESSOR AND ITS PERIPHERAL
23                ;       EQUIPMENT.
24                ;
25                ;       THIS TEST IS PROVIDED IN THREE VERSIONS:
26                ;       SHORT,LONG,PERIPHERAL.
27                ;
28                ;       ALL THREE VERSIONS WILL PERMIT RUNNING OF
29                ;       SLAVE IOPS CONTAINING IPMORT IF THEY
30                ;       WERE LOADED BY DTOS.(HIOP DIRECTORY)
31                ;
32                ;       THE SHORT VERSION ONLY INCLUDES THOSE TESTS
33                ;       THAT APPLY TO THE CPU, MEMORY, FLOATING POINT,
34                ;       MAP,DEMAND PAGING,CHARACTER/COMMERCIAL,
35                ;       ARRAY PROCESSOR,DCU-50/200,I/O TESTER,
36                ;       RTC,TTY.
37                ;
```

```
!0004  SEMRT
01                ;2.     HARDWARE CONFIGURATIONS SUPPORTED
02                ;
03                ;2.1    MINIMUM MACHINE REQUIREMENTS
04                ; 2.1.1 ECLIPSE M600,S130,S250,C350,C150 PROCESSOR
05                ; 2.1.2 32K OF READ WRITE MEMORY (MUST
06                ;       BE CONTIGUOUS )
07                ; 2.1.3 TTY/CONSOLE(DEV. 10/11)
08                ;
09                ;2.2    OPTIONAL EQUIPMENT
10                ; 2.2.1  1024K OF READ/WRITE MEMORY(MUST BE CONTIGUOUS)
11                ; 2.2.2  ERROR CORRECTION
12                ; 2.2.3  MMPU1
13                ; 2.2.4  MMPU2 (DEMAND PAGING)
14                ; 2.2.5  CHARACTER/COMMERCIAL INSTRUCTION SET
15                ; 2.2.6  FLOATING POINT UNIT,FLT.PT.FUNCTIONS
16                ; 2.2.7  DCU 50 COMMUNICATIONS CONTROLLER
17                ; 2.2.8  I/O PROCESSORS(DEV. 60-67)
18                ; 2.2.9  ARRAY PROCESSOR(AP)
19                ; 2.2.10 6060 SERIES DISK (ANY/ALL DRIVES)
20                ; 2.2.11 6063/64 DISK (ANY/ALL DRIVES)
21                ; 2.2.13 MOVING HEAD DISK (ANY/ALL DRIVES)
22                ; 2.2.12 MAGNETIC TAPE (ANY/ALL DRIVES)
23                ; 2.2.13 LINE PRINTER (DCM/REG.,DEV.17 ONLY)
24                ; 2.2.14 REAL TIME CLOCK(DEV. 14)
25                ; 2.2.15 I/O TESTER(DEV. 0)
```

```
!0005  SEMRT
01                ;2.3    PREREQUISITES
02                ;2.3.1  SOFTWARE PREREQUISITES
03                ;       THE SYSTEM SHOULD BE CAPABLE
04                ;       OF RUNNING ALL INDIVIDUAL LOGIC AND
05                ;       RELIABILITY TESTS PERTAINING TO THE
06                ;       PROCESSOR AND ITS PERIPHERAL EQUIPMENT
07                ;       BEFORE ATTEMPTING TO RUN THIS TEST
08                ;NOTE:  ALTHOUGH THIS TEST MAY AT TIMES BE USEFUL
09                ;   IN DETERMING THE GO/NO GO STATUS OF AN
10                ;   UNKNOWN SYSTEM, IT IS RECOMMENDED THAT:
11                ;A. ALL OTHER DIAGNOSTICS BE RUN EVEN IN THE
12                ;   EVENT THAT THIS TEST FINDS NO PROBLEMS.
13                ;B. AN ATTEMPT BE MADE TO ISOLATE ANY PROBLEMS
14                ;   FOUND BY FIRST UTILIZING THE LOWER
15                ;   LEVEL TESTS FOR MORE CONSISE ERROR REPORTS.
16                ;2.3.2  SYSTEM SETUP
17                ;2.3.2.4 IF THE DCU IS TO BE RUN AT A DEVICE CODE OTHER THAN
18                ;       64 THEN ONE MEMORY LOCATION NEEDS TO BE UPDATED TO CONTAIN
19                ;       THE DCU 50/200 DEVICE CODE. SEE DCU TEST LOCATION DCUDV.
20        047670 PATCH=DCUDV
21                ;
22                ;3.       OPTIONAL STARTING ADDRESS
23                ;3.1      200   AUTO-SIZE AND GO START
24                ;3.2      202   MANUAL SELECT/DELETE TESTS START
25                ;3.3      204   RUN UP TO FIRST 32K UNMAPPED START
26                ;3.4      206   RESTART LAST TEST SELECTIONS
27                ;3.5      210   START IN THE ODT
28                ;3.6      212   AUTO-START WITH IPMONT
29                ;
30                ;       NOTE:   THIS PROGRAM CONTAINS AN OCTAL DEBUGGER
31                ;               TO ASSIST IN DETERMINING ERROR INFO
32                ;               OR TO BE UTILIZED TO PATCH THE TEST
33                ;               BEFORE STARTING.
34                ;
35                ;               IF STARTED FROM DTOS AND THIS PROGRAM
36                ;               VERSION IS ECLMORT L OR ECLMORT P THEN IT
37                ;               WILL AUTOMATICALLY START AT ADDRESS
38                ;               210 TO ALLOW SELECTION OF TESTS/PATCHING
39                ;               AND OR OPERATOR SETUP. TYPE P TO SELECT
40                ;               A 202 START,OR TYPE XXXR (WHERE XXX=200,
41                ;               202,204,212) FOR OTHER STARTS.
```

```
!0006  SEMRT
01                ;3.7    KEY ENTERED SWITCH OPTIONS(SWREG)
02                ;
03                ;  KEY 0        LOCKS THE SWPACKAGE INTO INPUT MODE
04                ;               ALLOWING SETUP OF THE CONTENTS OF "SWREG".
05                ;               TYPE A CARRAGE RETURN TO EXIT.
06                ;
07                ;       TYPING KEY'S 1 - 9,A - F SETS/RESETS SWREG BITS
08                ;       1 - 9,10 - 15 RESPECTIVELY. EACH KEY ENTRY
09                ;       COMPLEMENTS THE PREVIOUS STATE OF THE SWREG BIT.
10                ;
11                ; KEY   SWREG BIT       FUNCTION
12                ;-----  ---------       ----------------------------------------
13                ; 1     1 = 1           DOESN'T RELEASE OR ALLOW REASSIGNMENT
14                ;                       OF SCRATCH AREAS AFTER ERROR.
15                ;
16                ; 2     2 = 1           DELETE TTO TYPEOUTS
17                ;
18                ; 3     3 = 1           INCREASE CHANCE OF ENTERING THE TTY
19                ;                       AND THE LPT TESTS.
20                ;
21                ; 4     4               TYPING A 4 WILL CAUSE THE ELAPSED RUN
22                ;                       TIME AND ACCUMULATED ERRORS
23                ;                       TO BE TYPED ON THE TTY.
24                ;                       (NOTE: A RTC MUST EXIST)
25                ;
26                ; 5     5 = 1           DIRECT ERROR AND RUNTIME TYPEOUTS
27                ;                       ALSO TO THE LINE PRINTER .
28                ;
29                ; 6     6 = 1           THE ERROR ROUTINE WILL PAUSE AFTER
30                ;                       EACH PHASE OF AN ERROR TYPEOUT.
31                ;                       TYPE A CR KEY ON DEVICE TTI TO PROCEED.
32                ;
33                ; 7     7               TYPING A 7 WILL CAUSE
34                ;                       INDIVIDUAL RUN STATISTICS OF
35                ;                       EACH TEST TO BE LISTED.
36                ;
37                ; 9     9 = 1           INHIBIT MMPU2 PAGE AND BREAKPOINT FAULTS
38                ;
39                ; A     10 = 1          INHIBIT MMPU2 BREAKPOINT FAULTS
40                ;
41                ; B     11 = 1          DELETE CONTEXT BLK FROM TYPEOUTS
42                ;
43                ; D     13 = 1          INCREASE NUMBER OF MMPU2 FAULTS
44                ;                       TAKEN IF DEMAND PAGING TEST ENABLED
45                ;                       AND SWREG 9 = 0 .
46                ;
47                ; E     14 = 1          PROGRAMMED WAIT WHILE PRINTING MMPU2
48                ;                       CONTEXT TABLE AND UDT MAP TABLE OUTPUTS.
49                ;
50                ;  NOTE:  (C) = CONTROL KEY
51                ;
52                ; I                     SET INPUT MODE TO IOP. ALL
53                ;                       FOLLOWING KEY INPUTS ARE DIRECTED
54                ;                       TO IOP.
55                ;
56                ; H                     SET INPUT MODE TO LOCAL.(CLEARS
57                ;                       IOP INPUT MODE).
58                ;
59                ; M                     PRINTS THE CURRENT CONTENNTS OF SWREG.
60                ;
```

```
 0007  SEMRT
01              ;(C)D                   RESTARTS THE PROGRAM AT LOC 202
02              ;                       AFTER RESETTING SWREG TO ALL 0'S
03              ;
04              ;(C)R                   RESTARTS THE PROGRAM AT LOC 202
05              ;                       WITHOUT DISTURBING SWREG.
06              ;
07              ; EXAMPLE:
08              ;       TO DELETE ERROR TYPEOUTS AND LOOP ON FAILING
09              ;       SCRATCH AREAS TYPE 0,1,2, AND CR.
```

```
10008  SEMRT
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 TESTS SELECTED
08              ;       210 TO START IN ODT
09              ;       212 TO AUTO START WITH IPMORT
10              ;
11              ;NOTE:  THE TEST MUST HAVE BEEN STARTED
12              ;       AND RUN FOR A FEW SECONDS
13              ;       BEFORE ADRS 206 MAY BE UTILIZED
14              ;4.3      PRESS START
15              ;         PROCESSOR WILL TYPE:
16              ;       ECLMORT-(S,L,P) (MULTI-PROGRAMMING REL. TEST XXX VER)
17              ;       TOTAL #1K'S=XXX(DECIMAL) MMPU1 OR NO MMPU1
18              ;       IOP EXISTS(IF FOUND)
19              ;       PROGRAM RUN LIST
20              ;       PROG# DESCRIPTION
21              ;4.4      IF START WAS 200 (OR 206) THE LIST OF
22              ;       PROGRAMS TO BE RUN CONCURRENTLY WILL
23              ;       THEN BE LISTED AND THE TEST SYSTEM
24              ;       WILL AUTO START
25              ;4.5      IF START WAS 202 LINKER WILL
26              ;       PAUSE AT THE END OF EACH TEST
27              ;       DESCRIPTION AND WAIT FOR KEYBOARD
28              ;       INPUT.  TYPING IN A SPACE WILL
29              ;       ENABLE THAT TEST TO BE RUN.
30              ;       TYPING IN ANY OTHER CHARACTER WILL
31              ;       DELETE THAT TEST FROM BEING RUN
32              ;4.6      IF THE STARTING ADDRESS WAS 204 THE LINKER
33              ;       WILL SIZE MEMORY WITHOUT UTILIZING
34              ;       OR EVEN LOOKING FOR THE MMPU1 OPTION
35              ;       AND THEN PROCEED AS IN STARTING AT ADRS. 202
36              ;       WITH THE MMPU1 NONEXISTENT.
37              ;4.7    IF THE ODT WAS STARTED (ADDRESS 210) AND "P"
38              ;       IS TYPED THE TEST WILL BE STARTED AS IF
39              ;       IT WAS STARTED AT 202.
40              ;4.8    OPTION SET UP
41              ;4.8.1  THE PROGRAM WILL PAUSE AFTER PRINTING
42              ;       "ENTER OPTIONS,CR TO CONTINUE" AND ALLOW
43              ;       KEY ENTRY OF SWITCH REG. OPTIONS. TYPE
44              ;       A "CR" KEY TO START TESTS.
45              ;4.8.2  IF AN AUTOSTART ADDRESS WASN'T USED
46              ;       AND AN IOP WAS FOUND DURING SIZING
47              ;       THE OPERATOR WILL BE ASKED TO SPECIFY THE
48              ;       IOP STARTING ADDRESS FOR EACH IOP.
49              ;       ENTER VALUE FROM 200 TO 214 TO START,0 TO DELETE.
50              ;       TERMINATE INPUT WITH A CR.(VALUES OUTSIDE
51              ;       THIS RANGE WILL NOT BE ACCEPTED.)
52              ;       (SEE IPMORT LISTING FOR STARTING ADDRESS
53              ;       DESCRIPTIONS).
54              ;4.9    IF STARTED THE IOPS WILL OUTPUT THEIR HEADER
55              ;       AND RUN LISTS AT THIS TIME AND IF NOT AN AUTO
56              ;       START WILL ALLOW TEST SELECTION/DELETION
57              ;       AND OPTION SETUP.
58              ;       EACH TEST WHICH HAS SIZING INFORMATION
59              ;       WILL PRINT THAT INFORMATION THE FIRST TIME
60              ;       THE TEST IS SELECTED FOR EXECUTION.
```

```
 0009  SEMRT
01                ;5.       ERROR DESCRIPTIONS
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                ;
19                ;         NOTE:    ALL TYPEOUTS ORIGINATING IN A
20                ;                  SLAVE IOP WILL BE PRE-FACED BY THE
21                ;                  IOP NUMBER.
22                ;
23                ;         ERROR TYPEOUTS INCLUDE:
24                ;
25                ;5.1.1    PROGRAM # AND NAME AT TIME OF ERROR
26                ;5.1.2    THE CURRENT CONTENT OF THE USER STACK
27                ;         IN UP TO 2 SETS OF 5 VARIABLES.
28                ;         EACH SET OF VARIABLES (1 STATE BLOCK) INCLUDE
29                ;         AC0 TO AC3 AND THE PC WITH CARRY.
30                ;NOTE:  THE 2 STATE BLOCKS TYPED ARE THE FIRST
31                ;         AND LAST 5 ENTRIES ON THE USER STACK.
32                ;         IF THERE ARE SAY SEVEN ENTRIES ON THE STACK
33                ;         THESE STATE BLOCKS WILL OVERLAP AND FOR THIS
34                ;         REASON THE START ADRS OF EACH STATE BLOCK
35                ;         IS INCLUDED IN THE TYPEOUT
36                ;5.1.3    LOGICAL SCRATCH AND DATA CHANNEL LIMITS
37                ;5.1.4    MEMORY ALLOCATION TABLE
38                ;         PHYSICAL 1K PAGE# +LOGICAL ADDRESS
39                ;         +REL LOG. ADDR(IF RELOCATED AND REMAPPED)
40                ;5.1.5    CONTINUATION INFORMATION IN GROUPS
41                ;         OF 3 MEMORY LOCATIONS PERTINENT TO
42                ;         THE INDIVIDUAL TEST THAT FAILED
43                ;
44                ;5.1.6    THE CPU TESTS THAT RELOCATE/REMAP WILL
45                ;IN THEIR ERROR TYPEOUTS:
46                ;ST.LA    START/ERROR (RES.)
47                ;XXXXXX YYYYYY  ZZZZZZ
48                ;
49                ;ST.LA    THE LOGICAL START OF THE RELOCATED TEST LOOP
50                ;XXXXXX (I.E. THE LAST LCALL SETUL)
51                ;
52                ;START    THIS NUMBER INDICATES WHERE THE RESIDENT COPY
53                ;YYYYYY OF THE TEST LOOP MAY BE FOUND IN THE LISTING
54                ;
55                ;ERROR    THIS NUMBER INDICATES WHERE IN THE RESIDENT
56                ;ZZZZZZ COPY OF THE LISTING THE ERROR CALL MAY BE FOUND
57                ;         (FOR SOME VALIDITY TRAP ERRORS THIS NUMBER
58                ;         MAY NOT APPEAR TO BE VALID.)
```

```
 0010  SEMRT
01                ;5.1.8  DEMAND PAGING DATA FOR PAGE AND
02                ;         BREAKPOINT FAULTS.
03                ;
04                ; IF THE DEMAND PAGING TEST IS SELECTED AND THE SW9/SW10
05                ;SELECTION IS IN THE ENABLE STATE(=0), THEN ANY ERRORS
06                ;INCURRED IN THE CPU TESTS WHICH REMAP AND RELOCATE COULD
07                ;BE DUE TO INTERREACTIONS WITH THE MMPU2. TO DETERMINE
08                ;IF THIS IS THE CASE EITHER SET SW9 TO THE INHIBIT
09                ;STATE (=1) OR RESELECT TESTS WITH THE DEMAND PAGING
10                ;TEST DELETED.
11                ;
12                ;5.1.8.1 CONTEXT BLOCK INFORMATION RETREIVAL
13                ;
14                ; IF A TEST WAS TAKING PAGE/BREAKPOINT FAULTS
15                ; AND SW11 IS IN THE ENABLE STATE(=0)
16                ; THE CONTEXT BLOCK INFORMATION WILL BE PRINTED
17                ;FOLLOWING THE MEMORY ALLOCATION TABLE. ALSO THE CONTEXT
18                ;BLOCK INFO. CAN BE OBTAINED BY ENTERING THE ODT AND TYPING
19                ; A "C".(SEE SECTION 7. FOR ODT DESCRIPTION)
20                ;
21                ; NOTE: THIS TABLE IS 38 WORDS LONG BY FIVE WORDS WIDE
22                ;         USE A HARDCOPY OUTPUT DEVICE TO RECOVER ALL
23                ;         THE ERROR OUTPUT.
24                ;
25                ;5.1.8.2          CONTEXT BLOCK OUTPUT ORGANIZATION
26                ;
27                ; THE CONTEXT BLOCK DATA IS PRINTED IN FIVE VERTICAL
28                ;COLUMNS, THE MOST RECENT BEING ON THE LEFT AND THE
29                ;LEAST CURRENT ON THE RIGHT. USUALLY THE TWO LEFT
30                ;COLUMNS WILL BE EXACT COPIES BECAUSE BEFORE THE DPOP
31                ;BY THE FAULT HANDLER THE CONTEXT INFORMATION IS
32                ;SHIFTED TOWARD THE RIGHT. IF THE TWO LEFT COLUMNS
33                ;AREN'T IDENTICAL THEN THE LEFT MOST COLUMN WAS
34                ;CAUSED BY A VALIDITY MMPU1 FAULT, NOT A MMPU2 FAULT.
35                ;
36                ;         NOTE:    THE FPU INFORMATION IN THE CONTEXT BLK
37                ;                  IS ONLY VALID IF WORD 15 BIT 0 IS A"1"
38                ;
39                ;5.1.8.3          CONTEXT BLOCK FORMAT
40                ;
41                ; WORD 1           VALIDITY WORD # 1
42                ;      2                "     "   # 2
43                ;      3-5         GR0 - GR2
44                ;      6-9         JSR0 - JSR3
45                ;     10           STATUS(BITS0-3)+RAS
46                ;     11           CPU STATE(BITS0-7)+BREAKCONTROL
47                ;     12           I.R.
48                ;     13           CPU BIT COUNTER
49                ;     14           PAGE
50                ;     15           F.P.+BREAKPOINT ADDR.
51                ;     16-19        AC0 - AC3
52                ;     20           P.C.
53                ;     21           FLOATING POINT STATUS REG BITS 0-15
54                ;     22            "        "     "        "  "   16-31
55                ;     23-26        FPAC0
56                ;     27-30        FPAC1
57                ;     31-34        FPAC2
58                ;     35-38        FPAC3
```

```
10011  SEMRT
01              ;5.1.9  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              ;5.1.10 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 WORD
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.
41              ;
```

```
10012  SEMRT
01              ;5.1.11 BMC CHANNEL
02              ;
03              ;       IN THE NORMAL M600 CONFIGURATION THE 6060/61
04              ;       AND 6063/64 DISKS ARE CONNECTED TO THE BMC
05              ;       CHANNEL. IF PROBLEMS ARE FOUND TO EXIST
06              ;       ON EITHER DISK THE DISK COULD BE RECONFIGURED
07              ;       TO USE THE DCH INSTEAD. THIS WILL HELP ISOLATE
08              ;       WHETHER THE PROBLEM IS THE BMC OR DISK.
09              ;       THE DISK TESTS WILL SIZE THE CHANNEL
10              ;       IN USE UPON INITIALIZATION. ALSO DISK
11              ;       TESTS WHICH USE THE BMC WILL OUTPUT THE
12              ;       BMC STATUS WORD IN AC1 UPON A STATUS ERROR
13              ;
14              ;***********************************************
15              ;  BMC CHANNEL STATUS WORD
16              ;***********************************************
17              ;  BITS         MEANING
18              ;   0           ERROR
19              ;   1           I/O DIRECTION
20              ;   2           TWO STEP MODE
21              ;   3           VALIDITY PROTECT ERROR
22              ;   4-6 N/A
23              ;   7           ADDR PARITY ERROR
24              ;   8           DATA PARITY ERROR
25              ;*********************************************
26              ;5.2    ERROR ANALYSIS
27              ;       DUE TO THE INTERACTIVE NATURE OF
28              ;       THE TESTS INVOLVED, A SERIES OF
29              ;       ERROR TYPEOUTS WILL PROBABLY BE
30              ;       REQUIRED FOR ANALYSIS BEFORE A
31              ;       PROBLEM WILL BE ISOLATED.
32              ;       A RESTART AT 202 AND DELETION OF ALL
33              ;       BUT THE TEST THAT ORIGINALLY
34              ;       FAILED MAY HELP TO ISOLATE
35              ;       INTERACTIVE PROBLEMS AS FOLLOWS:
36              ;
37              ;5.2.1  IF THE TEST RUNS BY ITSELF THE PROBLEM
38              ;       IS INTERACTIVE-RE-ENABLE ONE OTHER TEST AT
39              ;       A TIME TO DETERMINE WHICH ONE IS THE PROBLEM.
40              ;       IF THE TEST DOES NOT RUN BY ITSELF
41              ;       RESORT TO SIMILAR BUT LOWER LEVEL TESTS
42              ;       FOR ISOLATION
43              ;
44              ;5.3    PERTINENT MEMORY LOC'S TYPED
45              ;
46              ;5.3.1  CHECKERBOARD RAN
47              ;
48              ;       THE AC'S AT ERROR WILL INDICATE:
49              ;       GOOD DATA- BAD DATA-LOGICAL ADDRESS
50              ;
51              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
52              ;       CB.TK   TEST COUNTER
53              ;       0       GENERATE CHECKERBOARD
54              ;       1       DISTURB PASS
55              ;       2       CHECK PATTERN
56              ;       3       CHECKSUM THE # OF -1'S IN PATTERN
57              ;       CB.LC   STARTING LOGICAL ADDRESS OF "BEGIN"
58              ;               RELOCATED TO SCRATCH
59              ;       CB.SE   AC3 AT ERROR CALL
```

```
!0013  SEMRT
01              ;5.3.2  ARITHMETIC 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              ;       AT.LC   STARTING ADDRESS OF ARITH IN SCRATCH
08              ;       AT.LO   LOW LIMIT OF SCRATCH AREA AFTER IT IS
09              ;               REMAPPED FOR EXECUTION
10              ;       AT.LA   AT.LC IN RELATION TO AT.LO
11              ;               (LOGICAL START OF ARITH AFTER REMAPPING)
12              ;       THE LAST THREE RANDOM NUMBERS GENERATED
13              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
14              ;
15              ;5.3.3  EIS/MRI TEST
16              ;
17              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
18              ;       TIME OF ERROR DETECTION
19              ;
20              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
21              ;       EI.TK   SEE DISCUSSION OF TEST FOR THE
22              ;               SEQUENCE BEING EXECUTED
23              ;       EI.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
24              ;               REMAPPED FOR EXECUTION
25              ;       EI.LA   LOGICAL START OF TEST AFTER REMAP
26              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
27              ;
28              ;5.3.4  FLT PT AND FLT PT FUNCTIONS TEST
29              ;
30              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
31              ;       TIME OF ERROR DETECTION
32              ;
33              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
34              ;       FP.LC   START ADRS. OF FLT. PT. TEST
35              ;               FOR RELOCATION
36              ;       FP.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
37              ;               REMAPPED FOR EXECUTION
38              ;       FP.LA   LOGICAL START OF TEST AFTER REMAP
39              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
40              ;
41              ;5.3.5  EXTENDED ADDRESSING TEST TEST
42              ;
43              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
44              ;       TIME OF ERROR DETECTION
45              ;
46              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
47              ;       EA.LC   START ADRS. OF TEST
48              ;               FOR RELOCATION
49              ;       EA.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
50              ;               REMAPPED FOR EXECUTION
51              ;       EA.LA   LOGICAL START OF TEST AFTER REMAP
52              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
```

```
!0014  SEMRT
01              ;5.3.6  LEF/ERROR 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              ;       LF.LC   START ADRS. OF TEST
08              ;               FOR RELOCATION
09              ;       LF.LO   LOW LIMIT OF SCRATCH AREA AFTER IT WAS
10              ;               REMAPPED FOR EXECUTION
11              ;       LF.LA   LOGICAL START OF TEST AFTER REMAP
12              ;       (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)
13              ;
14              ;5.3.7  ADDRESS TEST
15              ;
16              ;       THE AC'S 0 1 AND 2 WILL
17              ;        BE TYPED AS THEY
18              ;       WERE AT THE TIME OF ERROR
19              ;       AC0 WILL CONTAIN THE ADDRESS THAT COMPARE WILL
20              ;       STOP AT (AC0<AC2=COMPARE IS TOP TO BOTTOM)
21              ;       AC1 WILL CONTAIN THE DATA FOUND TO BE INCORRECT
22              ;       AC2 WILL CONTAIN THE ADDRESS OR COMPLIMENT
23              ;               OF THE ADDRESS THAT FAILED
24              ;       AC3 WILL CONTAIN THE CURRENT RANDOM OFFSET
25              ;       IN ADDITION THE FOLLOWING LOC.'S WILL BE TYPED:
26              ;       A.TSK   TEST COUNTER
27              ;       0       ADRS. TO ADRS LOADED LOW TO HIGH
28              ;       2       COM. ADRS TO ADRS LOW TO HIGH
29              ;       4       ADRS. TO ADRS LOADED HIGH TO LOW
30              ;       6       COM. ADRS TO ADRS LOADED HIGH TO LOW
31              ;       10      ADRS TO ADRS LOADED LOW TO HIGH
32              ;       1,5, AND 11 COMPARE EACH ADRS. TO EQUAL ITSELF
33              ;       3 AND 7 COMPARE EACH ADRS. TO EQUAL ITS COM.
34              ;       AD.ST   STARTING ADDRESS OF TEST IN CORE
35              ;       AD.SS   ADRS. (AC3) OF ERROR CALL
36              ;
```

```
!0015  SEMRT
01                ;5.3.9  ARRAY PROCESSOR TEST
02                ;
03                ;       THE AC'S AT THE TIME OF ERROR WILL BE PRINTED
04                ;       ALONG WITH THE FOLLOWING:
05                ;
06                ;       AP.LC   START OF TEST BEFORE REMAPPING
07                ;       AP.LO   SCRATCH LOW AFTER REMAPPING
08                ;       AP.LA   LOGICAL START OF TEST AFTER REMAPPING
09                ;               (SEE DISCUSSION OF ST.LA,ETC. AT PARAG. 5.1.6)
10                ;
11                ;       ALSO THERE ARE THREE DIFFERENT ERROR PRINTOUTS
12                ;THAT MAY ARISE DURING THE AP TESTS:
13                ;
14                ;1.     WHEN COMPARING REAL ARRAYS THE ROUTINE .RCMP
15                ;IS USED. THE FIRST 3 BAD COMPARISONS ARE PRINTED OUT.
16                ;PRINTED ON THE FIRST LINE, OF THIS AP OUTPUT SECTION,
17                ;IS THE INDEX OF THE ERROR AND THE GOOD PIECE OF DATA.
18                ;THE NEXT LINE CONTAINS THE BAD DATA ADDRESS AND THE BAD
19                ;PIECE OF DATA. THE ADDRESS OF THE GOOD DATA CAN BE FOUND
20                ;ON THE LISTING, IN THE SUBROUTINE CALL.
21                ;       AFTER A MAXIMUM OF THREE ERROR ARE PRINTED A
22                ;FINAL LINE WILL CONTAIN THE FOLLOWING:
23                ;       A)THE NUMBER OF ELEMENTS COMPARED = N
24                ;       B)THE GOOD ARRAY INDEX VALUE = IJ
25                ;       C)THE BAD ARRAY INDEX VALUE = IK
26                ;       C)THE TOTAL # OF ERRORS IN THAT COMPARISON
27                ;
28                ;2.     THE INTEGER ARRAY COMPARE ROUTINE, .ICMP, HAS
29                ;A VERY SIMILAR OUTPUT TO .RCMP.THE EXCEPTIONS ARE:
30                ;       A)LINE ONE CONTAINS THE INDEX OF THE BAD DATA AND
31                ;THE ADDRESS OF THE BAD DATA.
32                ;       B)THE NEXT LINE CONTAINS THE GOOD DATA AND THE
33                ;BAD DATA.
34                ;
35                ;       ********************************************************
36                ;       *
37                ;       *       NOTE:  THIS TEST ASSUMES THAT THE ARRAY
38                ;       *              PROCESSOR MICRO-CODE HAS BEEN
39                ;       *              PREVIOUSLY LOADED
40                ;       *
41                ;       ********************************************************
42                ;5.3.10 WCS TEST
43                ;
44                ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE TIME
45                ;       OF ERROR DETECTION .
46                ;
47                ;
48                ;       IN ADDITION THE FOLLOWING LOCTIONS ARE TYPED:
49                ;
50                ;       WS.LC   STARTING ADDRESS OF WCS TEST IN SCRATCH
51                ;       WS.LO   LOW LIMIT OF SCRATCH AREA AFTER IT IS
52                ;               REMAPPED FOR EXECUTION.
53                ;       WS.LA   WS.LC IN RELATION TO WS.LO
54                ;               (LOGICAL START OF WCS TEST AFTER REMAPPING)
55                ;       THE LAST THREE NUMBERS GENERATED
56                ;       (SEE DISCUSSION OF ST.LA,ETC. AT PARAGRAPH X.X.X)
57                ;
```

```
!0016  SEMRT
01                ;5.3.11 COMMERCIAL INSTRUCTION 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                ;       CM.LC   START ADRS OF TEST (RELOCATED)
08                ;       CM.LO   LOW LIMIT OF SCRATCH AREA AFTER IT
09                ;               WAS REMMAPED FOR EXECUTION
10                ;       CM.LA   LOGICAL START OF TEST AFTER REMAP
11                ;       (SEE DISCUSSION OF ST.LA,ETC. AT PARA. 5.1.6)
12                ;
13                ;       A COMMERCIAL FAULT WILL CAUSE AN ADDITIONAL
14                ;       STACK ENTRY TO BE PUSHED WHICH WILL CONTAIN
15                ;       THE AC'S AT THE TIME OF ERROR PLUS THE PC
16                ;       OF THE RELOCATED INSTRUCTION CAUSING THE FAULT.
17                ;        THE SECOND STACK ENTRY WILL BE PUSHED BY
18                ;       THE ERROR HANDLER AND AC1 WILL CONTAIN
19                ;       THE FAULT CODE AS DETERMINED BY THE HARDWARE.
20                ;
21                ;*****************************************
22                ;       COMMERCIAL FAULT CODES:
23                ;*****************************************
24                ;       CODE    MEANING
25                ;       0       "EDIT" INSTRUCTION TRIED TO
26                ;               PROCESS AN INVALID OP CODE.
27                ;       1       INVALID DATA TYPE
28                ;       2       INVALID SIGN
29                ;       3       INVALID DIGIT OR CHARACTER
30                ;       4       NUMBER OUT OF RANGE USING
31                ;               LDI OR STI INSTRUCTION
32                ;
33                ;5.3.12 DCU-50/200 TEST
34                ;
35                ;       THIS IS AN ARITHMETIC TEST PERFORMED BY THE
36                ;       DCU-50/200 USING THE DATA CHANNEL.
37                ;
38                ;       THE AC'S AT THE TIME OF ERROR DETECTION WILL
39                ;       BE TYPED .
40                ;       IN ADDITION THE FOLLOWING DATA IS TYPED:
41                ;       RANDOM DATA AC0,AC1,AC2
42                ;       DCLOP   LOGICAL START OF LOOP
43                ;       DCLPK   LOOP COUNT
44                ;       DCLER   LOGICAL ERR ADDR
45                ;       DC.LA   LOGICAL START OF TEST
46                ;       DC.LP   LISTING START OF LOOP
47                ;       ERROR   LISTING ADDR OF ERROR
```

```
10017  SEMRT
01              ;5.3.16 SC MEMORY TEST
02              ;
03              ;       THIS IS AN ISZ/DSZ TEST FOR SC-MEMORIES.
04              ;
05              ;       THE AC'S AT ERROR WILL INDICATE:
06              ;       AC0     AC1     AC2
07              ;       ACTUAL EXPECTED LOGICAL ADDRESS
08              ;
09              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
10              ;       MM.TK   ERROR NUMBER:
11              ;       0       PATTERN STORING ERROR(SHD BE =1)
12              ;       1       LOCATION NOT =1 BEFORE DOING ISZ
13              ;       2       ISZ DION'T SKIP
14              ;       3       LOCATION NOT EQUAL TO 0 AFTER ISZ
15              ;       4       DSZ SKIP ERROR
16              ;       5       DSZ TEST-LOCATION NOT =1 AFTER DSZ
17              ;       6       SAME AS 1, EXCEPT TESTING IN REV DIRECTION
18              ;       7       SAME AS 2, EXCEPT   "      "  "     "
19              ;       10      SAME AS 3, EXCEPT   "    "       "       "
20              ;       MM.SE   INSTRUCTION ADDRESS FOLLOWING ERROR CALL
21              ;       LOCATION  ADDRESS OF FAILING LOCATION(LOGICAL)
```

```
10018  SEMRT
01              ;5.3.19 I/O TESTER TEST
02              ;
03              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE TIME
04              ;       OF ERROR DETECTION.
05              ;
06              ;       IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
07              ;
08              ;       IO.OC   COMMAND BEING PERFORMED
09              ;       IO.BK   # OF BLK'S READ/WRITTEN
10              ;       B#ERR   BLK # IN ERROR
11              ;       ERRDST  LOGICAL ADDR. OF LOC CONTAINING BAD DATA
12              ;       IOCST   BEGINNING DCH ADDRESS
13              ;
14              ;       NOTE: IF THE OPERATION BEING PERFORMED WAS A
15              ;       WRITE( BIT 1 OF IO.OC = 0) AND IF
16              ;       A) AC2 IS WITHIN SCRLO/HI LIMITS
17              ;               THEN ERROR IS DUE TO CHECK OF OUTPUT
18              ;               BUFFER AREA AFTER A WRITE COMMAND.
19              ;       B) AC2 IS NOT WITHIN SCRLO/HI LIMITS
20              ;               THEN ERROR IS AN XOR RESULT ERROR.
21              ;
```

```
10019  SEMRT
01              ;5.4    SPECIAL CASE ERROR TYPEOUTS
02              ;
03              ;5.4.1  POWER FAIL INERRUPT
04              ;       UPON DETECTION OF A POWER FAIL INTERRUPT
05              ;       THE LOGICAL ADRS. OF THE P.C. AT INTERRUPT
06              ;       WILL BE SAVED.
07              ;       IF AUTO-RESTART IS ENABLED OR THE POWER
08              ;       FAIL WAS ONLY MOMENTARY, THE TEST WILL RE-
09              ;       START AS IN A START AT 206 AFTER TYPING
10              ;POWER FAIL @XXXXXX (WHERE XXXXXX IS THE PC AT INTR.)
11              ;
12              ;5.4.2  ILLEGAL SUPERVISOR CALL
13              ;
14              ;       SUPERVISOR CALLS IN MULTI-PROGRAMMING ARE OF THE
15              ;       FORMAT:
16              ;               SYC     3,3     ;SUPERVISOR CALL
17              ;               JSR     @XXXX   ;SUBROUTINE CALL
18              ;
19              ;       WHERE XXXX IS A PAGE 0 ADDRESS WITHIN
20              ;       SPECIFIED MEMORY LIMITS.
21              ;
22              ;       THESE CALLS IN THIS PROGRAM ALLOW A MAPPED
23              ;       TEST TO MAKE CALLS TO SUBROUTINES CONTAINED
24              ;       IN THE LINKER AND RETURN. THE JSR IS EXECUTED
25              ;       IN UNMAPPED SPACE BY THE CALL HANDLER.
26              ;       THERE ARE SEVERAL CASES OF ILLEGAL SUPER-CALL'S
27              ;       1.) IF THE LOCATION INDICATED IS 000001
28              ;         THEN THE PROGRAM DETECTED THAT FOR SOME REASON
29              ;         LOCATION 000000 WAS EXECUTED. THE ILLEGAL
30              ;         SUPERCALL WAS FORCED BY THE ERROR HANDLER
31              ;         TO INDICATE WHICH TEST WAS EXECUTING AT THE
32              ;         TIME OF ERROR AND ITS ASSOCIATED INFORMATION.
33              ;       2.) IF OTHER THAN LOCATION 000001 THEN
34              ;         THE SUPERCALL HANDLER WAS ENTERED AND UPON
35              ;         CHECKING THE LOCATION FOLLOWING THE SUPER-
36              ;         CALL THERE WASN'T A VALID "JSR @XXXX".
37              ;         THIS COULD HAPPEN IN SEVERAL WAYS:
38              ;        A.) THE PROGRAM COULD HAVE JUMPED INTO
39              ;          A DATA AREA WHICH CONTAINED A WORD
40              ;          RECOGNIZED AS A SUPERCALL. IN THIS
41              ;          CASE AC3 OF THE SECOND ENTRY OF THE
42              ;          STACK USUALLY ISN'T 177510(SYC 3,3).
43              ;        B.) THE JSR @XXXX WORD WAS OVER WRITTEN
44              ;          BY ANOTHER TEST.
45              ;        C.) THE 177510(SYC 3,3) OR THE JSR @XXXX
46              ;          WAS IN A BAD MEMORY AREA CAUSING THE
47              ;          WORDS TO DROP OR ADD BITS.
48              ;        D.) THE JSR @XXXX WAS IN A DIFFERENT MAP
49              ;          PAGE WHOSE CONTENTS ISN'T CORRECT.
50              ;        E.) SOME OTHER REASON ??
51              ;
52              ;  IN ANY CASE THE FOLLOWING INFORMATION IS TYPED:
53              ;       THE STACK SHOULD CONTAIN TWO BLOCKS.
54              ;       THE FIRST IS PUSHED BY THE ILLEGAL CALL.
55              ;        (IF SYC 3,3 THEN HEADER IS CORRECT)
56              ;       THE SECOND IS BUILT BY THE ILLEGAL CALL
57              ;       HANDLER AND INCLUDES THE FOLLOWING INFO.
58              ;       AC0     ILLEGAL CALL IN LOC. XXXXX
59              ;       AC3     CONTENTS OF LOC. XXXXX-1
60              ;               (SHOULD = SYC 3,3 OR 177510)
```

```
 0020  SEMRT
01              ;
02              ;5.4.3  ERROR TRAP
03              ;       AN I/O,WRITE,DEFER OR VALIDITY TRAP
04              ;       OCCURED THAT WAS NOT FORCED BY ANY TEST
05              ;       THE STATE BLOCK TYPED IS THAT PUSHED AS A
06              ;       RESULT OF THE TRAP
07              ;       IN ADDITION THE FOLLOWING ARE TYPED:
08              ;
09              ;*********************************************
10              ; DIA           MAP STATUS AS FOLLOWS
11              ;*********************************************
12              ;       BIT 1   EXT. FAULT
13              ;       BIT 2   I/O PROTECT TRAP
14              ;       BIT 3   WRITE PROTECT TRAP
15              ;       BIT 4   DEFER PROTECT TRAP
16              ;       BIT 5   SINGLE REF. TRAP
17              ;       BIT 6-8 FORMAT BITS
18              ;       BIT 9   LEF MODE
19              ;       BIT 10  I/O PROTECT ENABLE
20              ;       BIT 11  WRITE PROTECT ENABLE
21              ;       BIT 12  DEFER PROTECT ENABLE
22              ;       BIT 13  A/B USER SEL.
23              ;       BIT 14  DCH ENABLE
24              ;       BIT 15  USER MODE
25              ;*****************************************
26              ;
27              ; INSTR-THE INSTRUCTION WHICH IS LOCATED AT
28              ;       PC-1 AND SHOULD BE CAUSEING THE TRAP.
29              ;
30              ;*****************************************
31              ; DIC   MAP STATUS "C" AS FOLLOWS
32              ;*****************************************
33              ;       BIT 0   WRITE PROTECT BIT
34              ;       BIT 1-3 FORMAT BITS
35              ;       BIT 5   EXT PAGE
36              ;       BITS 6-15 PHYSICAL BLK #
37              ;                 (1777=VALIDTY PROTECT)
```

```
10021  SEMRT
01                ;
02                ;5.4.4  INTERRUPT WAIT ELAPSED
03                ;       THE PERIPHERAL DEVICE ASSOCIATED WITH THE
04                ;       PROG. NUMBER TYPED HAS NOT RESPONDED WITH
05                ;       A PROGRAM INTERRUPT FOR AN EXTENDED
06                ;       PERIOD OF TIME. THE 2ND NUMBER TYPED
07                ;       SHOULD POINT AT THE INTERRUPT HANDLER
08                ;       FOR THE DEVICE THAT FAILED
09                ;
10                ;5.4.5  ERROR CORRECTION INTERRUPTS
11                ;       IF ERCC IS ENABLED THE TYPE OUTS THAT
12                ;       OCCUR WILL MORE LOGICALLY BE DUE TO
13                ;       ERROR CORRECTION INTERRUPTS THAN
14                ;       TO DATA FAILURES DETECTED BY
15                ;       ANY OF THE GOOD/BAD COMPARES
16                ;       AGAIN, THE INITIAL TYPEOUT WILL CONSIST
17                ;       OF THE FOUR AC'S:
18                ;       AC0=    NOT USED
19                ;       AC1= DIB FROM ERCC OPTION
20                ;               BITS 0 TO 4 ARE CORRECTION CODE
21                ;               BITS 13-15 ARE THE COMPLEMENT
22                ;               OF PHYSICAL EXT.ADDR BITS BNK2,XPA0-1
23                ;       AC2=DIA,COMPLEMENT OF THE FOLLOWING
24                ;               BIT 0 IS THE EXT.PHYSICAL ADDRESS BIT XPA2
25                ;               BITS 1-3 ARE THE PHYS.ADDR. BITS 1-3
26                ;               BITS 12-15 ARE THE PHYS. BITS PA12-PA15
27                ;       AC3= ADDRESS(PC) OF THE INTERRUPT
28                ;               (REFER TO LISTING)
29                ;       THE CORRECTION CODE IS RIGHT JUSTIFIED
30                ;       AND TYPED
31                ;       THE TOTAL NUMBER OF ERROR CORRECTION
32                ;       INTERRUPTS THAT OCCURED DURING THE LAST
33                ;       TEST LOOP ARE TYPED
34                ;
35                ;       NOTE:   THIS VERSION OF ERROR CORRECT ALLOWS
36                ;               IDENTIFICATION OF ONLY WHICH 4K MODULE
37                ;               OF MEMORY HAS FAILED AND WHICH INTERLEAVING
38                ;               LEAF IT IS IN.
39                ;
40                ;       EXAMPLE:IF,
41                ;               AC1     AC2
42                ;               130017  120013
43                ;               THEN,
44                ;       BNK0-2,XPA0-2=0
45                ;
46                ;       PA1-3=101   GIVES A PHYSICAL ADDRESS
47                ;               RANGE OF 50000 TO 57777
48                ;       PA 12-15=0100 WHICH GIVES INTERLEAVE BOARD #4
49                ;               IN 8 WAY SCHEME
```

```
10022  SEMRT
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
```

```
10023  SEMRT
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 1024K AND BUILDS A 64 WORD
17                ;      BIT MAP OF EXISTING CONTIGUOUS
18                ;      MEMORY
19                ;
20                ;   5. THE EXIST MAP IS MOVED TO THE
21                ;      AVAILABLE MAP AND EACH BIT
22                ;      CORRESPONDING TO 1K OF UTILIZED
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. THE OPERATOR CAN THEN
40                ;      SELECT OR DELETE SPECIFIC TESTS,
41                ;       START THE IOP, AND ENTER KEY OPTIONS
42                ;      IF START WAS 202 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.)
48                ;
```

```
10024  SEMRT
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                ;
07                ;          1. LINKER RANDOMLY SELECTS ONE OF
08                ;             THE INDIVIDUAL TESTS UNTIL IT
09                ;             FINDS ONE THAT IS NOT WAITING
10                ;             FOR INTERRUPT (WAIT IS BIT 0 OF
11                ;             THE THIRD WORD IN TEST=1) AND THAT
12                ;             THE NEXT RANDOM NUMBER FALLS WITHIN
13                ;             ITS ENTER LIMITS
14                ;
15                ;          2. IF THE MAP OPTION EXISTS, ALL LOGICAL PAGES
16                ;             EXCEPT PAGE 0 ARE ACCESS PROTECTED WITH
17                ;             THE PHYSICAL AREA OF THE SELECTED TEST
18                ;             MAPPED TO ITSELF AND ANY ASSIGNED
19                ;             SCRATCH AREA MAPPED TO START AT 1K
20                ;             ABOVE THE TEST, MEMORY LOCATIONS SCRLO
21                ;             AND SCRHI (SCRATCH LOW AND HIGH) ARE
22                ;             SET TO INDICATE THE LIMITS OF
23                ;             THE SCRATCH AREA AVAILABLE TO THE TEST.
24                ;
25                ;          3. DATA CHANNEL LIMITS (DCHLO AND DCHHI)
26                ;             ARE CALCULATED AND ENTERED(IF USED)
27                ;
28                ;          4. BMC LIMITS ARE CALCULATED AND SETUP.
29                ;            (IF USED)
30                ;          5. THE SELECTED TEST IS ENTERED AT
31                ;             ITS SPECIFIED EXECUTE ENTRY POINT
32                ;
```

```
01
02              ;6.4    INDIVIDUAL TEST DESCRIPTIONS
03              ;
04              ;6.4.1 DEMAND PAGING TEST
05              ;
06              ;       THIS TEST DOESN'T RUN IF THE OPTION TO RUN UNMAPPED
07              ; WAS SELECTED. IF ENABLED THIS TEST CAUSES THE CPU TESTS
08              ; WHICH REMAP AND RELOCATE TO TAKE PAGE FAULTS AND BREAKPOINT
09              ; FAULTS IN THEIR THEIR NORMAL PROCESS OF TESTING.
10              ; TO ACCOMENDATE THE FAULTING PROCESS THE TESTS ARE RESTRICTED
11              ; TO USER "B" AND THE TESTS WHICH REMAP WILL DO SO IN USER "A".
12              ;       INSTEAD OF SETTING UP THE MAP FOR USER "A" AS REQUIRED
13              ; BY THE TEST WHICH IS REMAPPING, ALL PAGES ARE SETUP TO PAGE
14              ; FAULT. THE TEST IS ENTERED BY FAULTING INTO THE STARTING
15              ; PAGE. AS EACH FAULT OCCURS THE MICRO-CODE PUSHES A CONTEXT
16              ; BLOCK AND GIVES CONTROL TO THE SUPERVISOR'S FAULT HANDLER.
17              ; THE FAULT HANDLER SETS UP THE MAP AS REQUIRED BY THE TEST
18              ; FOR THAT 1K AND INVALIDATES THE LAST 1K USED. THE FAULT
19              ; HANDLER THEN SAVES THE CONTEXT BLOCK IN A PUSH DOWN STACK.
20              ; AFTER WHICH IT RESTARTS THE TEST VIA A UPOP INSTRUCTION.
21              ;       TESTS WHICH ARE TAKING FAULTS WILL OUTPUT THE
22              ; CONTENTS OF THE PUSH DOWN STACK UPON OCCURING AN ERROR
23              ; IF SW11 IS NOT SET.
```



```
01              ;6.4.2  ERROR CORRECTION
02              ;
03              ;  THE ERROR CORRECTION OPTION IN MULTIPROGRAMMING
04              ; NORMALLY ENABLES ERROR CORRECTION IN MODE 3 (CORRECT
05              ;AND INTERRUPT).  IF AN ERROR CORRECTION INTERRUPT OCCURS THIS
06              ;TEST SAVES ALL INFORMATION AVAILABLE PERTAINING TO THE FIRST
07              ;ERROR FOR LATER TYPEOUT. BETWEEN THE TIME OF THE ERROR CORRECTION
08              ;INTERRUPT AND COMPLETION OF THE TYPEOUT ERROR CORRECTION IS
09              ;RE-ENABLED IN "MODE 3". HOWEVER, UNTIL THE TYPEOUT IS COMPLETED,
10              ;ERROR CORRECTION INTERRUPTS ARE ONLY COUNTED
11              ;NOTE:  THE TYPEOUT DOES NOT OCCUR IMMEDIATELLY AT THE TIME
12              ;       OF INTERRUPT, IT IS INITIATED AFTER
13              ;       COMPLETION OF THE CURRENT PASS OF THE TEST THAT
14              ;       WAS RUNNING WHEN THE INTERRUPT OCCURRED. ALSO,
15              ;       IF THE ADDRESS OF THE ERROR WAS OUTSIDE OF THE
16              ;       SCRATCH AND RESIDENT AREAS OF THE TEST THAT WAS
17              ;       RUNNING, ONE SHOULD SUSPECT THAT THE ERROR
18              ;       COULD HAVE OCCURRED DURING A DATA CHANNEL.
19              ;6.4.3  CHECKERBOARD RAN
20              ;
21              ;THIS MEMORY CHECKER BOARD TEST IS A SUBSET OF OTHER MEMORY
22              ;CHECKERBOARDS.  A COMPLETE TEST OF AN AVAILABLE SCRATCH
23              ;AREA IS COMPRISED OF THE FOLLOWING SEQUENCE:
24              ;
25              ;CB.TK=0        ;REQUEST 1 TO 32K OF SCRATCH, RANDOMLY RE-
26              ;               ;LOCATE THE EXECUTE PORTION OF CHECKERBOARD
27              ;               ;INTO SCRATCH AND GENERATE THE CHECKERBOARD
28              ;               ;PATTERN
29              ;
30              ;CB.TK=1        ;DISTURB PASS-COMPLIMENT A SINGLE BIT IN EACH
31              ;               ;OF THE FIRST 16 WORDS OF SCRATCH, SHUFFLE THESE
32              ;               ;WORDS 16 TIMES SUCH THAT THEY END UP IN THEIR
33              ;               ;ORIGINAL POSITION, RE-COMPLIMENT THE SINGLE
34              ;               ;BIT IN EACH WORD-PROCEED WITH EACH GROUP OF
35              ;               ;16 WORDS UNTIL ALL MEMORY HAS BEEN EXERCISED.
36              ;
37              ;CB.TK=2        ;CHECK PASS-COMPARE EACH WORD IN SCRATCH WITH
38              ;               ;THE PATTERN EXPECTED
39              ;
40              ;CB.TK=3        ;FAST CHECKSUM MEMORY TO ENSURE THAT ALL DATA
41              ;               ;IS INTACT (RETURNS TO CHECK PASS IF CHECK-
42              ;               ;SUM DOES NOT AGREE.)
```

```
10027  SEMRT
01                ;6.4.4  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 VIA 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 STA 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)
```

```
10028  SEMRT
01                ;6.4.5  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
```

```
10029  SEMRT
01                ;6.4.6  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                ;
14                ;6.4.7  FLOATING POINT TEST
15                ;
16                ;THE MULTIPROGRAMMING FLOATING POINT TEST IS SIMILAR IN OP-
17                ;ERATION TO THE ARITHMETIC TEST.  THE FPU TEST DATA BUFFER IS
18                ;LOCATED RANDOMLY WITHIN 32 WORDS IN THE CENTER OF THE EX-
19                ;ECUTE PORTION OF THE FLT PT TEST.  THE FLT. PT. NUMBERS PRO-
20                ;CESSED BY THIS TEST ARE FIXED (NOT RANDOM) AND CAN BE FOUND
21                ;SPECIFIED ON THE LISTING FOR EACH TEST.
22                ;
23                ;6.4.8  FLOATING POINT FUNCTIONS TEST
24                ;
25                ;FLOATING POINT FUNCTIONS BEING TESTED INCLUDE:  SQUARE ROOT,
26                ;EXPONENTIAL, NATURAL LOGARITHM, SINE, COSINE, AND POLYNOMIAL
27                ;(SINGLE AND DOUBLE PRECISION FOR ALL TESTS).  THE TEST IS RANDOMLY
28                ;RELOCATED TO THE ASSIGNED SCRATCH AREA WHERE IT IS EXECUTED.
29                ;
30                ;6.4.9  LEF/ERROR TEST
31                ;
32                ;THE LEF MODE - ERROR TEST IN MULTIPROGRAMMING RELIABILITY IS
33                ;ONLY RUN IF AN MMPU EXISTS.  THE LEF PORTION OF THIS TEST VERIFIES
34                ;THAT THE LEF ENABLE ON THE MMPU FUNCTIONS IN ALL ADDRESSING
35                ;MODES.  THE ERROR PORTION OF THE TEST VERIFIES THAT THE WRITE, I/O
36                ;DEFER AND VALIDITY PROTECT FEATURES OF THE MMPU FUNCTION
37                ;CORRECTLY.  SCRATCH AND EXECUTION ARE TREATED AS IN ARITH-
38                ;METIC TEST.
39                ;
40                ;6.4.10 EXTENDED ADDRESSING TEST
41                ;
42                ;THE EXTENDED ADDRESSING TEST
43                ;VERIFIES THE CORRECT OPERATION OF THE DOUBLE LENGTH
44                ;INSTRUCTIONS,THE IMMEDIATE MODE DOUBLE LENGTH,
45                ;DISPA,CLM,MSP,HLV AND ELEF.
46                ;SCRATCH AREA IS TREATED AS INTHE EIS/MRI TEST
47                ;WITH TEST EXECUTION SIMILAR TO THE ARITH,FLT PT.
48                ;AND LEF/ERROR TESTS
```

```
10030  SEMRT
01                ;6.4.11 ARRAY PROCESSOR TEST
02                ;
03                ;       THIS TEST, VERIFIES THE LDR AND
04                ;STR INSTRUCTIONS AS A PAIR. USING LDR, AN ARRAY OF 128.
05                ;SEQUENTIAL, FLOATING POINT NUMBERS ARE LOADED TO AP RAM
06                ;FROM (M)AIN (M)EMORY BUFFER #1. ALL TESTS USE FIRST 1K
07                ;OF AP RAM. THEN USING STR INSTRUCTION, THE ARRAY IN RAM
08                ;IS STORED BACK TO MM IN BUFFER #2. BUFFERS #1 AND #2
09                ;ARE THEN COMPARED AND SHOULD BE THE SAME.
10                ;
11                ;       THE SRS AND ARS INSTRUCTIONS ARE ALSO TESTED AS
12                ;A PAIR. USING THE SEQUENTIAL, FLOATING POINT DATA AL-
13                ;READY LOADED IN RAM FROM PREVIOUS TEST AND THE ARS IN-
14                ;STRUCTION, 1.0 IS ADDED TO THIS ARRAY (CALL IT ARRAY A).
15                ;THE RESULT OF THIS OPERATION IS PUT RIGHT ABOVE ARRAY A
16                ;IN AP RAM (ARRAY B). THEN USING SRS INSTRUCTION 1.0 IS
17                ;SUBTRACTED FROM ARRAY B. THE RESULT OF THIS OPERATION
18                ;IS STILL IN ARRAY B. ARRAYS A AND B ARE THEN COMPARED
19                ;AND SHOULD BE THE SAME.
20                ;
21                ;       THE 3RD AND FINAL TEST IS MUCH MORE COMPLICATED.
22                ;THIS TEST FOURIER TRANSFORMS AN IMPULSE TO A STEP, THE
23                ;STEP IS THEN INVERSE TRANSFORMED BACK TO AN IMPULSE.
24                ;THE OUTPUT SHOULD EQUAL THE INPUT.
25                ;
26                ;       FIRST AN IMPULSE FUNCTION OF STRENGTH ONE, 128.
27                ;ELEMENTS LONG IS CREATED IN MM AS AN INTEGER ARRAY.
28                ;USING FLL INSTRUCTION, IT IS CONVERTED TO A FLOATING
29                ;POINT ARRAY AND LOADED TO AP RAM. THIS ARRAY IS FOURIER
30                ;TRANSFORMED USING FFTC INSTRUCTION. THE RESULT SHOULD
31                ;BE A STEP WHICH IS STORED IN A COMPLEX FORM. ALL 128
32                ;REAL PARTS ARE 1.0. ALL 128 IMAGINARY PARTS ARE 0.0.
33                ;THIS IS TESTED USING THE CMS EQ INSTRUCTION. IF THERE
34                ;IS AN ERROR HERE, THE ACS WILL BE PRINTED WITH AC0
35                ;CONTAINING RETURNED VALUE AND AC1 CONTAINING EXPECTED
36                ;VALUE. THE STEP IS NOW INVERSE TRANSFORMED USING THE
37                ;FFTC INSTRUCTION. THE RESULT SHOULD BE AN IMPULSE
38                ;WITH MAXIMUM HEIGHT 128. THIS MAX ELEMENT SHOULD BE
39                ;THE ZEROTH ELEMENT IN THE ARRAY. BOTH THESE CONDITIONS
40                ;ARE CHECKED USING THE MXP INSTRUCTION. ALL OTHER REAL
41                ;AND IMAGINARY ELEMENTS OF THIS IMPULSE ARRAY SHOULD
42                ;BE 0.0. THIS IS CHECKED VIA THE CMS EQ INSTRUCTION.
43                ;THIS IMPULSE FUNCTION ARRAY IN AP RAM IS THEN SCALED
44                ;DOWN TO STRENGTH ONE USING THE MRS INSTRUCTION. THEN
45                ;THE RESULTING REAL IMPULSE ARRAY IS INTEGERIZED BACK
46                ;TO MM USING THE FXS INSTRUCTION. THE RESULTING INTEGER
47                ;ARRAY IS CHECKED AGAINST THE ORIGINAL IMPULSE ARRAY.
48                ;THEY SHOULD BE THE SAME.
```

```
!0031  SEMRT
01                ;6.4.12 WCS TEST
02                ;
03                ;       THIS TEST CAN'T BE RUN WITH THE ARRAY
04                ;       PROCESSOR TEST.  IF THE ARRAY PROCESSOR
05                ;       IS INSTALLED BUT NOT TO BE TESTED, A
06                ;       PATCH IS REQUIRED TO ENABLE THE WCS
07                ;       TEST BEFORE IT CAN BE RUN.
08        035267 PATCH=   APSEL   ;PATCH TO 401
09                ;
10                ;       THE WCS TEST IS DIVIDED INTO FOUR MAIN
11                ;       AREAS WHICH ARE DESCRIBED BELOW.
12                ;
13                ;6.4.12.1       DEC1,DEC2 TEST
14                ;
15                ;       A DEC1 IS PERFORMED TO A RANDOM LOCATION
16                ;       IN WCS, FROM THERE A DEC2 IS PERFORMED
17                ;       TO ANOTHER RANDOM LOCATION IN WCS AND THEN
18                ;       AN EXIT OCCURS. THE LAST THREE RANDOM
19                ;       NUMBERS ARE USED AS FOLLOWS:
20                ;
21                ;       AC0     BITS 12-15 = ENTRY # =DEC1 ADDRESS
22                ;       AC1     BITS 1-4 = ACS ACD
23                ;       ADDRESS FOR DEC2 = 6-9,1-4, FROM XOP1
24                ;
25                ;6.4.12.2       FILE TEST
26                ;
27                ;       A RANDOM LOCATION IN THE FILE IS LOADED
28                ;       WITH RANDOM DATA. THE FILE IS THEN READ AND
29                ;       CHECKED.
30                ;
31                ;       AC0 = RANDOM DATA
32                ;       AC2 = RANDOM ADDRESS (BITS 8-15)
33                ;
34                ;6.4.12.3       ALU TEST
35                ;
36                ;       THE ALU TEST IS A GROUP OF MICRO-INSTRUCTIONS
37                ;       WHICH ARE LOADED WITH A RANDOM STARTING LOCATION
38                ;       IN WCS AND WHICH EXERCISE ALL THE ALU FUNCTIONS
39                ;       ALONG WITH FUNCTIONS IN THE FOLLOWING FIELDS:
40                ;       A-PORT,A-INPUT,B-PORT,SHIFT,LOAD, AND CARRY.
41                ;
42                ;       THE THREE RANDOM #'S IN AC0,AC1,AC2 ARE USED AS
43                ;       OPERANDS AND THE RESULTS FROM EXECUTION
44                ;       OF THE WCS SEQUENCE ARE COMPARED WITH SIMULATED
45                ;       RESULTS.
46                ;
47                ;       THE LAST THREE RANDOM NUMBERS ARE USED AS FOLLOWS:
48                ;       AC0     BITS 8-15       =FIRST ADDRESS IN WCS
49                ;       AC2     BITS 12-15      =ENTRY # (0-17)
50                ;       AC0                     =OPERAND 1
51                ;       AC1                     =OPERAND 2
52                ;       AC2                     =OPERAND 3
53                ;
54                ;6.4.12.4       MEMORY TEST
55                ;
56                ;       THE WCS MEMORY TEST PRIMARILY EXERCISES THE
57                ;       MA AND MBUS FIELDS.
58                ;
59                ;       THE TEST SEQUENCE IN WCS GENERATES TWO
60                ;       NEW WORDS DEPENDENT UPON THE
```

```
 0032  SEMRT
01                ;       ORIGINAL RANDOM NUMBER IN TWO WORK LOCATIONS.
02                ;       THE MAIN PROGRAM SIMULATES THE PROCEDURE AND
03                ;       COMPARES THE ACTUAL WITH THE EXPECTED RESULTS.
04                ;
05                ;       THE LAST THREE RANDOM NUMBERS ARE USED AS FOLLOWS:
06                ;
07                ;       AC0     BITS 8-15       =FIRST ADDRESS IN WCS
08                ;       AC2     BITS 12-15      =ENTRY # (0-17)
09                ;       AC0                     =ORIGINAL MR0
10                ;       AC1                     =ORIGINAL MR1
11                ;
```

```
10033  SEMRT
01                ;6.4.13 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 TOGHETHER
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 TOGHETHER. 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 LOAD
19                ;ONE AT A TIME BY THE ELDB INSTRUCTION BACK INTO THE ACCU
20                ;THE RANDOM NUMBER IS HENCED RECONSTRUCTED AND COMPARED WITH
21                ;THE ORIGINAL FOR ERROR . THIS EXERCISE CHECKS THAT THE EL
22                ;ESTB INSTRUCTIONS DO NOT DISTURB THE ADJACENT BYTES.
23                ;ALL FOUR ADDRESSING MODES ARE TESTED SEPARATELY.
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 TO
32                ;THE CTR AND CMT INSTR. ARE EXERCISED SEPARATELY
33                ;AND THEIR TRASLATION TABLE IS LOCATED RANDOMLY IN THE
34                ;BUFFER. SINCE THE CTR INSTRUCTION REQUIRES A TRANSLATION
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 OF 256 WORDS. IN THIS CASE THE LOWER HAL
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 INTO 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 SEPARATELY.
52                ; THE EDIT INSTRUCTION IS USED TO CONVERT INTEGERS
53                ;FROM ONE FORMAT TO THE OTHER. THREE TESTS ARE
54                ;WRITTEN IN WHICH ALL EDIT OP CODES ARE EXERCISED.
```

```
10034  SEMRT
01                ;6.4.14 DCU-50/200 TEST
02                ;
03                ;THE MULTI-PROGRAMMING DCU-50/200 TEST RUNS
04                ;AN ARITHMETIC TEST VIA THE DATA CHANNEL INTO
05                ;THE HOST MEMORY.
06                ;THE DCU-50/200 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                ;
14                ;6.4.15 SC MEMORY TEST
15                ;
16                ;THIS MEMORY TEST DOES A READ/MODIFY/WRITE TO THE AVAILABLE
17                ;SCRATCH AREA USING THE "ISZ AND DSZ" INSTRUCTION.
18                ;THE ISZ/DSZ TEST IS MOVED INTO SCRATCH AT EITHER END
19                ;AND TESTED AREA IS THE UNOCCUPIED SCRATCH AREA.
20                ;THE TEST IS BROKEN INTO THE
21                ;FOLLOWING CHECKS:
22                ;
23                ;MM.TK= 0        WRITE INTO EACH MEMORY LOCATION A MINUS
24                ;                ONE AND VERIFY EACH LOCATION.
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.
```

```
!0035  SEMRT
01                ;6.4.21 REAL TIME CLOCK
02                ;
03                ;THE REAL TIME CLOCK IS RUN AT 100 HERTZ.  RUNTIME ALONG
04                ;WITH ACCUMULATED ERROR COUNT ARE PRINTED AT 5 MINUTES
05                ;15 MINUTES, 30 MINUTES AND EVERY 30  MINUTES OF RUNTIME
06                ;THEREAFTER.  THIS TYPEOUT ALSO OCCURS AFTER EVERY ERROR
07                ;TYPEOUT OR IF TTY KEY 4 IS TYPED.
08                ;* * * * * * * NOTE * * * * * * * * * * * * * * * * *
09                ;*  WHEN RUNNING THIS TEST WITH THE DCU-50/200 TEST AND/OR
10                ;* WITH IPMORT STARTED USING IT'S MAP, THE PRINTED TIME
11                ;* CAN LOSE TIME WITH RESPECT TO REAL RUN TIME. THIS IS
12                ;* DUE TO THE HEAVY CONCENTRATION OF DCH ACTIVITY CAUSED BY
13                ;* THESE DEVICES AS TESTED BY MULTI-PROGRAMMING REL. TEST
14                ;* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
15                ;
16                ;6.4.22 TELETYPE TEST
17                ;
18                ;THE TELETYPE TEST PRINTS A SINGLE LINE CONSISTING OF THE
19                ;CHARACTERS SPACE TO Z.  THE TEST WILL ALSO ECHO CHARACTERS
20                ;AS TYPED.
```

```
!0036  SEMRT
01                ;6.4.23 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 #6.
11                ;READ OR WRITE OPERATIONS ARE SELECTED RANDOMLY.
12                ;2 TO 4 K OF SCRATCH AREA ARE ASSIGNED TO THIS TEST.
13                ;
14                ;READ OPERATION:
15                ;THE I/O TESTER BUFFER IS LOADED WITH A RANDOM WORD
16                ;AND THE BLK SIZE IS SET UP FOR 64 WORDS. READ COMMAND IS
17                ;GIVEN AND REPEATED UNTIL ENOUGH BLKS TO FILL SCRATCH
18                ;ARE RECEIVED. THEN SCRATCH AREA IS CHECKED FOR
19                ;THE PROPER DATA WORD.
20                ;
21                ;WRITE OPERATION:
22                ;64 RANDOM WORDS ARE GENERATED AND THEN MOVED INTO
23                ;SCRATCH AREA REPEATEDLY UNTIL AREA IS FILLED.
24                ;WRITE COMMAND IS GIVEN AFTER SETTING BLK SIZE
25                ;TO 64 WORDS. THE IO TESTER INTERRUPTS AFTER EACH BLK
26                ;AND THE DATA BUFFER REGISTER IS READ AND STORED FOR
27                ;LATER CHECK. THIS IS REPEATED UNTIL ALL BLKS HAVE BEEN
28                ;RECEIVED. NEXT THE SCRATCH AREA IS CHECKED TO INSURE
29                ;IT WASN'T DISTURBED AND THEN A SOFTWARE XOR OF THE
30                ;64 WORDS OF DATA IS FORMED AND EACH RECEIVED WORD
31                ;IS COMPARED TO IT.
```

```
10037  SEMRT
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 AN UP-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              ;
55              ;       CHARACTERS TO SPECIFY IOP OR HOST INCLUDE:
56              ;       "I" "P" "R" "H"
57              ;       NOTE: A "R" OR "P" WHILE IN "I" MODE PLACES ODT IN
58              ;       HOST MODE.
59              ;       CHARACTERS USED TO MANIPULATE THE ECLIPSE MAP INCLUDE:
60              ;       "MM"  "MA"  "MB"  "MU"  "MT" "MNE" "MNL"
```

```
 0038  SEMRT
01              ;7.5 COMMANDS TO OPEN/CLOSE A LOCATION
02              ;  ~~~~~~~~~~~~~~~~~~~~~~
03              ;
04              ;       THE MEMORY LOCATION TO BE OPENED IS TYPEDOUT.
05              ;ADR/   OPEN THE LOCATION AND PRINT ITS CONTENTS
06              ;./     OPEN THE LOCATION CURRENTLY POINTED BY THE POINTER
07              ;        AND PRINIT ITS CONTENTS.
08              ;.+ADR/ ADD ADR TO THE POINTER, OPEN THE LOCATION AND
09              ;       PRINT ITS CONTENTS.
10              ;.-ADR/ SUBTRACT ADR FROM THE POINTER, OPEN THE LOCATION AND
11              ;       PRINT ADDR CONTENTS.
12              ; CR    CLOSE THE OPEN LOCATION WITH OR WITHOUT
13              ;       MODIFICATION OF ITS CONTENTS.
14              ; LF    CLOSE THE OPEN LOCATION WITH OR WITHOUT
15              ;       MODIFICATION OF ITS CONTENTS AND OPEN THE
16              ;       SUCCEDING LOCATION.
17              ;/      CLOSE THE OPEN LOCATION WITHOUT MODIFYING
18              ;       ITS CONTENTS AND OPEN THE CELL POINTED
19              ;       BY ITS CONTENTS
20              ;+ADR/  CLOSE THE OPEN LOCATION WITHOUT MODIFYING
21              ;       ITS CONTENTS AND OPEN THE LOCATION POINTED
22              ;       BY ITS CONTENTS+ADR
23              ;-ADR/  CLOSE THE OPEN LOCATION WITHOUT MODIFYING ITS
24              ;       CONTENTS AND OPEN THE LOCATION POINTED BY
25              ;       ITS CONTENTS-ADR.
26              ; ^     CLOSE THE CURRENT LOCATION AND OPEN ".-1"
27              ;
```

```
10039  SEMRT
01                ;7.6 OTHER COMMANDS
02                ;  --------------
03                ;
04                ;RUBOUT THE RUBOUT KEY IS USED TO DELETE ERRONEOUSLY TYPED
05                ;       DIGITS EACH TIME THE RUBOUT KEY IS PRESSED, THE RIGHT
06                ;       MOST DIGIT IS DELETED AND ECHOED ON THE TERMINAL.
07                ;       IF THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A CELL
08                ;       THEN IT ALLOWS THE MODIFICATION OF THE CONTENTS AS IF
09                ;       THEY WERE TYPED JUST BEFORE THE KEY WAS PRESSED.
10                ;
11                ; P     RESTART THE EXECUTION OF THE PROGRAM AT THE LOCATION
12                ;       FOLLOWING THE ONE WHICH CALLED ENTRY TO ODT. IF IN
13                ;       "I" MODE, THE IOP WILL BE STARTED AT THE SAVED PC
14                ;       AND THE MODE WILL BE CHANGED TO "H"(HOST).
15                ;
16                ; ADRR  START EXECUTION OF THE PROGRAM AT LOCATION ADR AFTER
17                ;       AN IO RESET. IF IN "I" MODE, THE SART WILL BE TO THE
18                ;       IOP AND THE MODE WILL BE SET TO "H"(HOST). USE ONLY
19                ;       THE STARTING ADDRESSES 200 TO 212 AS APPROPRIATE.
20                ;
21                ; I     STOPS THE IOP,SETS MODE TO IOP. ALL FOLLOWING COMMANDS
22                ;       WILL BE INTERPRETED AS IOP COMMANDS UNTIL EITHER
23                ;       " H,P,R " ARE TYPED.
24                ;
25                ; NA    TYPE THE CONTENTS OF SAVED ACCUMULATOR "N".
26                ;
27                ; K     KILL THE STRING TYPED SO FAR. ODT RESPONDS WITH A ? AND
28                ;       THE OPEN LOCATION IS CLOSED WITHOUT MODIFICATION.
29                ;
30                ; =     PRINT THE CURRENT ARGUMENT  (I.E.  TYPING ".=" WILL
31                ;       PRINT THE ADRS OF THE LAST OPENED LOCATION)
```

```
10040  SEMRT
01                ; 7.7   MAPPED COMMANDS
02                ;
03                ;       ALL MAPPED COMANDS MUST BE PRECEDED WITH A "M"
04                ;       COMMAND, I.E. "MM" "MA" "MB" "MC" "MT" "MNE" "MNL"
05                ;       ETC.
06                ;  ALL MAPPPED COMMANDS ARE VALID ONLY UNDER LOCAL MODE.
07                ;       IE. NOT TO AN IOP
08                ;
09                ; MM    SETS SWITCH SO THAT ALL MEMORY ACCESSES ARE
10                ;       MAPPED USING THE LAST USER'S SETUP UNTIL A "MU"
11                ;       "MA" OR "MB"  IS INPUTTED.
12                ;
13                ; MA    SETS SWITCH SO THAT ALL FURTHER MEMORY ACCESSES
14                ;       ARE MAPPED USING USER=A UNTIL A "MB" OR "MU" IS
15                ;       ENTERED.
16                ;
17                ; MB    SETS SWITCH SO THAT ALL FURTHER MEMORY ACCESSES
18                ;       ARE MAPPED USING USER=B UNTIL A "MA" OR "MU"
19                ;       IS ENTERED.
20                ;
21                ; MC    PRINTS THE CONTENTS OF THE LAST FIVE CONTEXT BLOCKS
22                ;       NOTE: FIRST TWO WILL BE IDENTICAL COPIES IF THE
23                ;       MOST RECENT PUSH WASN'T A MAP VIOLATION TRAP.
24                ;       THE "MC" COMMAND IS USEFUL ONLY ON MMPU2 TYPE
25                ;       PROCESSORS.
26                ;
27                ; MT    PRINTS THE CURRENT CONTENTS OF ALL MAP ENTRY TABLES
28                ;
29                ; MNE   PRINTS THE MAP ENTRY CORRESPONDING TO THE VALUE OF
30                ;       N TYPED, WHERE N IS THE LOGICAL PAGE ADDRESS.
31                ;       IF NO VALUE FOR N IS ENTERED THEN THE NEXT
32                ;       LOGICAL MAP ENTRY IS TYPED.
33                ;
34                ; MNL   SETS MAP LAST BLOCK TO VALUE ENTERED .
35                ;       OCTAL VALUES CAN RANGE FROM 0 TO 377.
36                ;       COMMANDS "A,B" CAN BE USED TO MODIFY FORMAT IF CALLED
37                ;       BEFORE "L" COMMAND.
38                ;
39                ; MU    CLEARS MAP MODE TO STOP MAPPING OF MEMORY ACCESSES.
40                ;
41                ; MNS   PRINTS THE CONTENTS OF THE BMC MAP WHICH CORRESPONDS
42                ;       TO THE LOGICAL PAGE ENTERED "N".
43                ;
44                ; NOTE: ENTERING OR EXITING ODT CLEARS MAPPED MODE.
45                ;
```

```
!0041  SEMRT
01                ;
02                ;7.8 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.
```