

IDENTIFICATION

PRODUCT CODE: MAINDEC-15-D1HA-D (D)
PRODUCT NAME: HOT MEMORY TEST, MK 11---HOTMEM
DATE CREATED: JANUARY 12, 1970
MAINTAINER: DIAGNOSTICS GROUP
AUTHOR: J.M. GRAETZ/J. RICHARDSON

21

1. ABSTRACT

This program tests the resistance of cores to being switched from the 1-to the 0-state during or after a rise in temperature. A core in the 1-state is heated by repeatedly addressing the core, using a JMP . instruction. In two seconds, the temperature of the core can rise as much as 15 C. (Cores in the 0-state are not affected). The repetitive read-write switching may cause an unstable core to revert to the 0-state. After the core has been heated, a half-select current passing through it may also cause the core to change state.

The program tests selected cores at (or as near as possible to) the corners of each array, in the middle of each edge, and at the center of the array. Provisions are made for error printouts and halts, scoping loops, and test repetition. Although intended primarily for extended memory, the test can be run on any PDP-15.

2. REQUIREMENTS

2.1 Equipment

Standard PDP-15 with any size core memory.

2.2 Storage

The program occupies locations 00000-000036, and 01000-01520. All other memory registers are cleared.

2.3 Preliminary Programs

None

3. LOADING PROCEDURE

The program is in absolute binary format.

- a. Place the program tape in the reader.
- b. Set the ADDRESS Switches to 17700.
- c. Set the BANK MODE Switch on a 1.
- d. Press I/O RESET, then READIN.

4. STARTING PROCEDURE

4.1

Control Switch Settings

AC SWITCH	STATE	MODE OR OPERATION
0	0	Normal
	1	Repeat current cycle
1	0	Halt after error print-out
	1	Don't halt after print-out
2	0	Print error data
	1	Do not print error data
3,4,5		Start testing in the 4K field specified by the octal value (0 to 7).
6,7,8		Octal number (0 to 7) of additional 4K extended memory fields.
16	0	Halt at end of program
	1	Repeat entire program
17	0	Go on to next field
	1	Repeat test on current field

4.2

Operator Action

- a. When the program has been loaded, remove the tape from the reader.
- b. Set the ADDRESS switches to 0200.
- c. Press I/O RESET, then START.
- d. A halt will occur with the PC=01006
- e. Set the control switches as desired, then press CONTINUE.

To re-start the program, follow steps a through d above.

5. OPERATING PROCEDURE

5.1 Switch Settings

See Section 4.1.

5.2 Operator Action

When the program is started, SW 0-2, 16, and 17, should be set to 0. If no errors occur, the program requires no intervention. To handle errors, several procedures are available.

a. Option 1- When an error first occurs, the data are printed and the program halts. To obtain information about possible failures in other cores, set SW 1 to 1, and press CONTINUE. At each error, the information is printed, and the test proceeds without stopping.

b. Option 2- To examine a signal trace at any given error site, set SW 0 and SW 2 to 1, then press CONTINUE. The cycle just executed will be repeated, whether or not the error occurs (making intermittent errors easily discernible). To observe the temperature-dependent behavior of the core, a 2 second cooling-off period is provided between repetitions of the cycle, during which the core is switched once every 18 microseconds.

The operator can examine a particular memory bank, or all banks in succession.

c. Normal Final Halt-C(PC)=01006. This is the same as the halt after starting, so that the user can proceed from a final halt as though starting the test.

6. ERRORS

6.1 Error Messages

At the first occurrence of an error, the following heading is printed:

HOT MEMORY TEST

CONTENTS LOCATION

Under "CONTENTS", the correct and actual contents of the tested register are printed, in that order. The absolute address of the register is printed under "LOCATION".

Normally, such an error printout occurs whenever a core is switched by a half-select current passing through it while the core is still hot. Sometimes, however, a core will

switch to the 0-state while it is being heated; if this happens, the following message is printed:

CORE SWITCHED WHILE HEATING

6.2 Error Stops and Recovery

<u>Tag</u>	<u>C(PC)</u>	<u>C(AC)</u>	<u>Data Printed.</u>	<u>Discussion</u>
SPIN +1	00031	I/O status	None	A device other than the KSR 33 teleprinter caused a program interrupt.
				Recovery: Disable the offending device (s) then press CONTINUE.
EHL	01144	n/a	CONTENTS and LOCATION of register	(1) A heated core was switched to the 0-state by a half-select current. (2) A core switched to the 0-state while being heated (Special message is printed).
				Recovery: Set the control switches as desired, then press CONTINUE.
EH2	01055	000000	None	The contents of SW3-5 are greater contents of SW6-8. The following message is printed: STARTING BLOCK TOO HIGH. RESET SW3-5.
				Recovery: Set SW 3-5 to be less than or equal to the setting of SW 6-8. Press CONTINUE.

6.3 Examples

Here is a typical error printout (the heading is printed only at the first occurrence of the error):

```
HOT MEMORY TEST
CONTENTS  LOCATION
605077
600077    24077
```

The register being tested is 24077 (that is, 4077 in memory bank 2). The correct contents should be 604077, which is the JMP . instruction. The actual contents are 600077, showing that bit 6 was switched to 0 incorrectly.

If the same error had occurred while core was being heated, the printout would have appeared as follows:

```
CORE SWITCHED WHILE HEATING
604077
600077      24077
```

7. RESTRICTIONS

None

8. MISCELLANEOUS

Execution Time: 62 seconds for each 4K memory bank.

9. DESCRIPTION

When a core in the 1-state is rapidly and repeatedly switched, it gets hot. If it gets too hot, a marginally-reliable core might spontaneously switch to the 0-state, or else it might be switched to 0 when a half-select current is passed through it (when another register on the same x- or y- line is addressed). This program is designed to catch such errors, examining critically-placed cores in as many bit arrays as is possible with the logic.

The rapid switching is effected by executing a JMP . instruction at the address under examination. All cores in the 1-state will get hot (0-state cores are not affected). By choosing the proper addresses, cores in every memory plane, and in almost every bit array (there are four arrays in a plane), can be tested under the most extreme conditions; in the center of the array, at the corners, and along the edges, where the cores are closest to the sense amps.

Table 1 lists the addresses used in the test. Because the operation code of JMP is 600000, bits 2 and 3 cannot be tested (they will always be 0). Likewise, bit 4 is not tested, because indirect addressing is not used. Bit 5 can be tested only in odd-numbered memory banks.

At the start of the test, the switches are examined to see how much memory is available, and in which bank the test is to start. Table pointers and indices are set up, and the entire test area is cleared to 000000.

A JMP . instruction is constructed and placed in the register being tested. The Teletype is selected (TLS + 10), the interrupt is enabled, and the program jumps to the tested register. The JMP . is repeatedly executed until a program interrupt occurs when the teletype flag is set; this occurs every 100 milliseconds. The flag is tallied, the Teletype reselected, and the JMP . is resumed. After 20 flags (2 seconds), the tested register is half-selected, and the contents of the tested register are compared against a standard.

If a bit changes while the core is being heated, the JMP . becomes either a JMP to another register, or a different instruction. In either case, the program counter is changed, and in most cases an empty register is addressed, causing the program to execute a CAL instruction which traps to location 00020. The program sequence starting in register 00021 causes the special message to be printed, followed by the error data.

When an error occurs, the switches are examined to see which options have been chosen. If SW 0 is set to 1, the program does not halt. Instead, a small time-wasting loop is executed to allow the heated cores to cool off for 2 seconds (the cores are switched every 18 microseconds, just to see if a core might change state while cooling off); the same register is then tested again.

The 30 addresses listed in Table 1 are tested for each memory bank provided. If SW 17 is set to 1, the test is repeated in the current bank. If SW 16 is set to 1, the entire test is repeated; otherwise, the program stops when all banks have been tested. The normal final halt is at PC=01006.

Table 1

Addresses of tested registers, for one 4K memory bank.

0037	0040	0077
0101	0102	0104
4001	4002	4004
7701	7702	7704
3700	3740	3777
4000	4037	4077
7700	7737	7740
0100	0240	0400
0140	0277	0440
0177		0477
7777		

Table 2

Addresses used for half-selecting tested registers

0000
4040
7777
7700
0137

```

        .TITLE HOT MEMORY TEST, MK II
        /
        /COPYRIGHT 1970, DIGITAL EQUIPMENT CORP,
        /MAYNARD, MASS.
        /
        /TEST OF CORE STABILITY DURING AND AFTER SELF-HEATING
        /
        .ARS
00001      .LOC      1
00001      600001    JMP      .
00002      777777    LAW      -1
00003      777777    LAW      -1
00004      777777    LAW      -1
00005      777777    LAW      -1
        /
00006      001100    IORET TEST      /'TEST' OR 'SCORE'
        /
00010      .LOC      10              /INDICES AND POINTERS
00010      000000    AXT      0      /TEXT POINTER
00011      000000    AXL      0      /CORE LOCATION TABLE
00012      000000    AX3      0      /MISC. INDEX
        /
00021      .LOC      21              /CAL CATCHER
00021      201534    LAC      (EM1-1  /SELF-HEATING ERROR
00022      041522    DAC      #ERAP   /ADDRESS OF TEXT
00023      601110    JMP      TEST2
        /
        /TELETYPE ROUTINE
        /
00024      441523    TTYR   ISZ      #FLAC   /COUNT FLAGS, DONE?
00025      600031    JMP      FXINT      /NO.
00026      620006    JMP*     IORET      /YES.
        /
00027      700314    SPIN   IORS      /SPURIOUS INTERRUPT
00030      740040    HLT
        /
00031      201531    FXINT  LAC      #SAC   /RESTORE AC
00032      703302    CAF
00033      700416    TLS+10
00034      700042    ION
00035      703344    ORR
00036      620000    JMP*     0
        /
00200      .LOC      200
00200      201535    LAC      (DAC #SAC   /SETUP PI HANDLER
00201      040001    DAC      1
00202      201536    LAC      (TSF
00203      040002    DAC      2
00204      201537    LAC      (JMP SPIN
00205      040003    DAC      3
00206      201540    LAC      (JMP TTYR
00207      040004    DAC      4
00210      601000    JMP      RGIN
        .EJECT

```

```

          /PROGRAM STARTS HERE
          /
01000    .LOC      1000
          /
01000    705514    BGIN  ISA+10    /DISABLE API IF PRESENT
01001    703302    CAF
01002    201541    LAC      (HFAD-1    /ADDRESS OF HEADING TEXT
01003    041525    DAC      #HSIG     /SET HEADER SIGNAL
01004    141522    DZM      #ERAP     /CLEAR MESSAGE POINTER
01005    750044    NFH   HLT:LAS     /NORMAL FINAL HALT
          /
          /INITIALIZATION
          /
01006    750004    INIT  LAS
01007    742010    RTL;   RAL
01010    740010
01011    501542    AND      (70000    /SR 6-8
01012    041521    DAC      #BANKS    /NO. OF BANKS
          /
          /CLEAR TEST AREAS
          /
01013    201543    LAC      (37      /CLEAR FROM 40...
01014    040012    DAC      AX3
01015    201544    LAC      (177     /...TO 177
01016    160012    DZM*    AX3
01017    540012    SAD      AX3     /DONE?
01020    751001    SKP:CLC    /YES. -1 TO AC
01021    601016    JMP      .-3     /NO.
01022    201545    LAC      (211     /CLEAR FROM 211
01023    040012    DAC      AX3
01024    201546    LAC      (777     /TO 777
01025    160012    DZM*    AX3
01026    540012    SAD      AX3
01027    751001    SKP:CLC
01030    601025    JMP      .-3
01031    341516    TAD      PLIM     /UPPER PROGRAM LIMIT
01032    040012    DAC      AX3     /CLEAR FROM THERE...
01033    201521    LAC      #BANKS
01034    341547    TAD      (7777    /...TO THE TOP OF MEMORY
01035    707702    FEM
01036    160012    DZM*    AX3
01037    540012    SAD      AX3     /DONE?
01040    601042    JMP      GU      /YES
01041    601036    JMP      .-3     /NO
          /
          .EJECT

```

HOT MEMORY TEST, MK II

```

/MAIN TEST SEQUENCE
/
01242 750004 GO LAS
01243 501542 AND (70000 /SR 3-5
01244 041533 DAC #XRITS /START IN THIS BANK
01245 740001 CMA
01246 341550 TAD (1 /2'S COMPLEMENT
01247 301521 ADD #BANKS /BANKS-XRITS
01250 740100 SMA /STARTING BANK TOO HIGH?
01251 601056 JMP G02 /NO.
01252 201551 LAC (EM2-1 /YES, PRINT WARNING.
01253 101255 PRINT
01254 750040 FH2 HLT!CLA
01255 601042 JMP GO
/
01256 201210 G02 LAC LTAB /RE-ENTRY TO REPEAT THE TEST
01257 040011 DAC AXL /LOCATION TABLE POINTER
/
01262 220011 G03 LAC* AXL /FETCH NEXT TEST ADDRESS
01261 741100 SPA /END OF TABLE?
01262 601173 JMP NEXB /YES.
01263 341533 TAD #XRITS /NO, ADD EXTENSION BITS...
01264 041524 DAC #HOP /...AND STORE FULL ADDRESS
01265 501552 AND (17777 /REDUCE TO 8K...
01266 241553 XOR (JMP /...AND BUILD INSTRUCTION
01267 041526 DAC #JTEM
/
01270 161524 HOTGO DZM* #HOP /RE-ENTRY FOR SCOPING.
01271 201526 LAC JTEM
01272 061524 DAC* #HOP /PLACE JMP INSTRUCTION
01273 777760 LAW -20 /FLAG COUNT
01274 041523 DAC #FLAG
01275 700042 ION
01276 700416 TLS+10 /INITIATE A FLAG
01277 621524 JMP* #HOP /GO TURN ON THE BURNER
/
/COME BACK HERE AFTER 2 SECONDS OF HEATING
/
TEST LAC* HS1 /THESE ADDRESSES HALF-SELECT
01100 221260 LAC* HS2 /ALL TESTED CORES
01101 221251 LAC* HS3
01102 221252 LAC* HS4
01103 221253 LAC* HS5
01104 221254 LAC* #HOP /NOW LOOK AT THE RECORD
01105 221524 LAC* JTEM /EVERYTHING OK?
01106 541526 SAD JTEM /YES.
01107 601150 JMP ION /NO.
/
. EJECT

```

```

/COME HERE AFTER SELF-DESTRUCT
/
01110 750004 TEST2 LAS /ENTRY FROM CAL TRAP
01111 742010 RTL /SW-2; PRINTING?
01112 751100 SPA:CLA /NO.
01113 601136 JMP TNG /YES, GET HEADFR ADDRESS
01114 201525 LAC #HSIG /PRINT HEADER?
01115 740200 SZA /YES.
01116 101255 PRINT

/SPECIAL MESSAGES
/
01117 201522 SPEM LAC #ERAP /ADDRESS OF MESSAGE
01120 740200 SZA /IS THERE ONE TO PRINT?
01121 101255 PRINT /YES.
01122 141522 DZM #ERAP

/PRINT THE DATA
/
01123 101365 CRLF
01124 201526 LAC JTEM /CORRECT CONTENTS
01125 101374 PROCT
01126 101365 CRLF
01127 707702 EEM
01130 221524 LAC* #HOP /ACTUAL CONTENTS
01131 101374 PROCT
01132 101352 TAB
01133 201524 LAC #HOP /ABSOLUTE ADDRESS
01134 101401 PROCS
01135 101365 CRLF

/
TNG LAS:CLL /HERE IF THERE WERE ERRORS
01137 740010 RAL /SW0 AND SW1
01140 741400 SZL /SCOPING?
01141 601154 JMP COOLIT /YES
01142 740100 SMA /NO, HALT?
01143 740040 FH1 HLT /YES.
01144 750004 LAS
01145 754100 SMA:CLA:CLL /SCOPING?
01146 601060 JMP G03 /NO, NEXT REGISTER.
01147 601070 JMP HOTGO /YES, DO THIS ONE AGAIN

/
TOK LAS /HERE IF NO ERRORS
01151 750100 SMA:CLA /SCOPING?
01152 601060 JMP G03 /NO.
01153 601154 JMP COOLIT /YES

.EJECT

```

```

/COOL-OFF PERIOD BETWEEN SCOPE LOOPS
/
01154 777754 COOL1T LAW -24
01155 041523 DAC FLAC /2-SECOND COUNT
01156 201554 LAC (SCORE /SET RETURN FROM BREAK
01157 040006 DAC IORET
01160 700042 ION
01161 700416 TLS+10
01162 221524 COOL2 LAC* #HOP /SWITCH THE CORES
01163 777774 LAW -4
01164 040012 DAC AX3 /18 USEC. DELAY
01165 440012 ISZ AX3
01166 601165 JMP .-1
01167 601162 JMP COOL2
/
01170 201555 SCORE LAC (TEST /COOL OFF DONE; RESTORE NORMAL...
01171 040006 DAC IORET /...BREAK RETURN
01172 601070 JMP HOTGO
/
/AFTER EACH 4K BANK IS TESTED, DO THIS:
/
01173 754004 NEXR LAS!CLL
01174 742020 RTR /SW 16-17 TO AC0,L
01175 741100 SPA /REPEAT THIS BANK?
01176 601056 JMP GO? /YES.
01177 201533 LAC #XBITS /NO. TEST FOR LAST BANK
01200 541521 SAD #BANKS /DONE ALL BANKS?
01201 601204 JMP NONE /YES.
01202 341556 TAD (10000 /NO. INCREMENT EXTENSION
01203 601044 JMP GO+2
/
01204 750400 NONE SNL!CLA /FINAL HALT?
01205 601005 JMP NFH /YES INDEEDY!
01206 141533 DZM #XBITS /NOPE. CLEAR EXTENSION BITS.
01207 601056 JMP GO? /AND GO ROUND AGAIN
/
.EJECT

```

/JMP LOCATION TABLE (AS 4K ADDRESSES)

```

/
LTAB .
01210 001210      0037;      0040;      0077
01211 000037
01212 000040
01213 000077
01214 000101      0101;      0102;      0104
01215 000102
01216 000104
01217 004001      4001;      4002;      4004
01220 004002
01221 004004
01222 007701      7701;      7702;      7704
01223 007702
01224 007704

/
01225 003700      3700;      3740;      3777
01226 003740
01227 003777
01230 004000      4000;      4037;      4077
01231 004037
01232 004077
01233 007700      7700;      7737;      7740
01234 007737
01235 007740
01236 000100      0100;      0400
01237 000400
01240 000140      0140;      0240;      0440
01241 000240
01242 000440
01243 000177      0177;      0277;      0477
01244 000277
01245 000477
01246 007777      7777;      777777
01247 777777

```

/HALF-SELECT ADDRESSES

```

/
HS1 0000
01250 000000      HS2 7777
01251 007777      HS3 4040
01252 004040      HS4 0137
01253 000137      HS5 7700
01254 007700
/

```

/CATCHES THE X=0 APPROXIMATION ADDRESSES

.EJECT

HOT MEMORY TEST, MK II

```

/TEXT PRINTER
/
PRINT.      0
            DAC      AXT
            DZM      #HSIG      /SUPPRESS FUTURE HEADERS
            LEM
PRINGO      LAC*      AXT      /FIRST PAIR-WORD
            DAC      #ASTEM
            CHAR     /POSITION FIRST CHARACTER...
            TESP
            CHAR     /...SECOND...
            TESP
            CHAR     /...PUT THE THIRD TOGETHER...
            AND      (170
            DAC      #AST3
            LAC*     AXT      /SECOND PAIR-WORD
            RCL;    PTL      /ADJUST
            DAC      ASTEM
            RAL     /3RD CHAR LOW-ORDER BITS
            AND      (7
            XOR      AST3     /ASSEMBLE THE PARTS
            TESP
            CHAR     /...FOURTH...
            TESP
            CHAR     /...AND FIFTH
            TESP
            JMP      PRINGO
/
/COME HERE FROM TESP WHEN TERMINATOR APPEARS
/
PROUT FEM
            JMP*     PRINT.
/
.EJECT

```

```

01255 000000
01256 040010
01257 141525
01260 707704
01261 220010
01262 041517
01263 101311
01264 101323
01265 101311
01266 101323
01267 101311
01270 501557
01271 041520
01272 220010
01273 744010
01274 740010
01275 041517
01276 740010
01277 501560
01300 241520
01301 101323
01302 101311
01303 101323
01304 101311
01305 101323
01306 601261
01307 707702
01310 621255

```

```

          /TEXT PRINTER SUBROUTINES
          /
          /1. POSITION CHARACTER
          /
01311    000000    CHAR. 0
01312    201517    LAC      #ASTEM
01313    744010    RCL
01314    742010    RTL;     RTL;     RTL
01315    742010
01316    742010
01317    041517    DAC      ASTEM      /NEXT CHAR. NOW LEFT-JUSTIFIED
01320    740010    RAL      /LAST BIT IN PLACE
01321    501544    AND      (177      /ISOLATE 7 BITS
01322    621311    JMP*     CHAR.
          /
          /2. TEST AND PRINT CHARACTER
          /
01323    000000    TESP. 0
01324    741200    SNA      /0? (ASCII FILLER)
01325    621323    JMP*     TESP.      /YES, IGNORE IT
01326    541544    SAD      (177      /TERMINATOR?
01327    601307    JMP      PROUT     /YES, GO WAY
01330    241561    XOR      (200      /NO, FINISH THE ASCII-CODE
01331    541545    SAD      (211      /TAB?
01332    601337    JMP      TES1      /YES.
01333    101341    TYPE     /NO.
01334    541562    SAD      (212      /CR?
01335    141532    DZM      TABCT      /YES, CLEAR TAB
01336    621323    JMP*     TESP.
01337    101352    TES1   TAB
01340    621323    JMP*     TESP.
          /
          /3. TYPE A CHARACTER
          /
01341    000000    TYPE. 0
01342    700002    IOF      /AVOID INTERRUPT
01343    700406    TLS
01344    700401    TSF
01345    601344    JMP      .-1
01346    700402    TCF
01347    700042    TON
01350    441532    ISZ      TABCT
01351    621341    JMP*     TYPE.
          /
          .EJECT

```

```

/4. TABULATOR
/
01352 000000  TAB. 0
01353 201532  LAC #TABCT /COUNT OF SPACES
01354 341563  TAB (-12 /REDUCE MODULO 10
01355 740100  SMA /NEGATIVE YET?
01356 601354  JMP .-2 /NO.
01357 040012  DAC AX3 /YES. STORE SPACE COUNT
01360 201564  LAC (240 /(SPACE)
01361 101341  TYPE
01362 440012  ISZ AX3 /DONE?
01363 601361  JMP .-2 /NO.
01364 621352  JMP* TAB. /YES
/
/5. CR-LF
/
01365 000000  CRLF. 0
01366 201565  LAC (215
01367 101341  TYPE
01370 201562  LAC (212
01371 101341  TYPE
01372 141532  DZM TARCT
01373 621365  JMP* CRLF.
/
101255 PRINT=JMS PRINT.
101311 CHAR=JMS CHAR.
101323 TESP=JMS TESP.
101341 TYPE=JMS TYPE.
101352 TAR=JMS TAB.
101365 CRLF=JMS CRLF.
/
.EJECT

```

```

/OCTAL PRINT SUBROUPINE
/
/ENTRY TO PRINT LEADING ZEROS
/
01374 000000 PROCT. 0
01375 041530 DAC #OCTEM
01376 201423 LAC OPS+3 /NOP
01377 101406 JMS OP1
01400 621374 JMP* PROCT.

/ENTRY TO REPLACE LEADING ZEROS WITH SPACES
/
01401 000000 PROCS. 0
01402 041530 DAC #OCTEM
01403 201566 LAC (SZA
01404 101406 JMS OP1
01405 621401 JMP* PROCS.

/PRINTER
/
01406 000000 OP1 0
01407 041420 DAC OPS /SET SWITCH
01410 777772 LAW -6
01411 041527 DAC #OCT /DIGIT COUNTER
01412 201530 OPGO LAC #OCTFM
01413 744010 RCL; RTL
01414 742010
01415 041530 DAC #OCTEM
01416 740010 RAL
01417 501560 AND (7
01420 740040 OPS XX /SZA OR NOP
01421 601427 JMP OPT
01422 441527 ISZ #OCT /LEADING ZERO. COUNT DIGITS
01423 740000 NOP
01424 201564 LAC (240 /TYPE A SPACE
01425 101341 TYPE
01426 601412 JMP OPGO

/
01427 241567 OPT XOR (260 /ASCII MAKER
01430 101341 TYPE
01431 201423 LAC OPS+3 /REPLACE SZA WITH NOP
01432 041420 DAC OPS
01433 441527 ISZ #OCT
01434 601412 JMP OPGO
01435 621406 JMP* OP1

/
PROCS=JMS PROCS.
PROCT=JMS PROCT.
/
.EJECT

```

```

      /MESSAGES
      /
      /HEADING TEXT
      /
01436 064251 HEAD .ASCII <15><12>'HOT MEMORY TEST'<15><12><12>
01437 047650
01440 202330
01441 546636
01442 512624
01443 052212
01444 516501
01445 505024
01446 416371 .ASCII 'CONTENTS'<11>'LOCATION'<15><12><177>
01447 652212
01450 472512
01451 304630
01452 476070
01453 152222
01454 476341
01455 505376

      /
      /SELF-DESTRUCT MESSAGE
      /
01456 064250 FM1 .ASCII <15><12>'CORE SWITCHED WHILE HEATING'<15><12><177>
01457 347644
01460 425012
01461 353622
01462 522071
01463 042610
01464 202571
01465 044630
01466 425011
01467 042602
01470 522231
01471 643432
01472 053760
01473 000000
01474 064252 FM2 .ASCII <15><12>'STARTING BANK TOO HIGH. RESET SW3-5.'<15><12><177>
01475 352202
01476 512511
01477 147216
01500 202050
01501 147226
01502 202511
01503 747500
01504 442230
01505 744134
01506 202450
01507 551612
01510 521012
01511 353546
01512 265525
01513 606424
01514 774020
01515 000000

```

```
01516 001570      PLIM / .SIZE
      000000      .END
      01534 001455 *L
      01535 041531 *L
      01536 700401 *L
      01537 600027 *L
      01540 600024 *L
      01541 001435 *L
      01542 070000 *L
      01543 000037 *L
      01544 000177 *L
      01545 000211 *L
      01546 000777 *L
      01547 007777 *L
      01550 000001 *L
      01551 001473 *L
      01552 017777 *L
      01553 600000 *L
      01554 001170 *L
      01555 001100 *L
      01556 010000 *L
      01557 000170 *L
      01560 000007 *L
      01561 000200 *L
      01562 000212 *L
      01563 777766 *L
      01564 000240 *L
      01565 000215 *L
      01566 740200 *L
      01567 000260 *L
```

SIZE=01570

NO ERROR LINES

ASTEM	01517
AST3	01520
AXL	00011
AXT	00010
AX3	00012
RANKS	01521
RGIN	01000
CHAR	101311
CHAR.	01311
CLOF	700004
CLON	700044
CLSF	700001
COOLIT	01154
COOL2	01162
CRLF	101365
CRLF.	01365
DONE	01204
EEM	707702
FH1	01143
FH2	01054
FM1	11456
FM2	01474
FRAP	01522
FXINT	00031
FLAC	01523
GO	01042
GO2	01056
GO3	01060
HEAD	01436
HOP	01524
HOTGO	01070
HSIG	01525
HS1	01250
HS2	01251
HS3	01252
HS4	01253
HS5	01254
INIT	01006
IORET	00006
JTEM	01526
KRR	700312
KSF	700301
LEM	707704
LTAB	01210
NEXR	01173
NEH	01005
OCCT	01527
OCTEM	01530
OPCO	01412
OPS	01420
OPT	01427
OP1	01406
PCF	700202
PLIM	01516
PRING0	01261

PRINT 101255
PRINT. 01255
PROCS 101401
PROCS. 01401
PROCT 101374
PROCT. 01374
PROUT 01307
PSA 700204
PSB 700244
PSF 700201
RCF 700102
RRR 700112
RSA 700104
RSB 700144
RSF 700101
SAC 01531
SCORE 01170
SPEM 01117
SPIN 00027
TAB 101352
TABCT 01532
TAB. 01352
TCF 700402
TESP 101323
TESP. 01323
TEST 01100
TEST2 01110
TES1 01337
TLS 700406
TNG 01136
TOK 01150
TSF 700401
TTYR 00024
TYPE 101341
TYPE. 01341
XBITS 01533
.EOT 00000

.EOT	00000
IORET	00006
AXT	00010
AXL	00011
AX3	00012
TTYR	00024
SPIN	00027
EXINT	00031
RGIN	01000
NFH	01005
INIT	01006
GO	01042
EH2	01054
G02	01056
G03	01060
HOTGO	01070
TEST	01100
TEST2	01110
SPEM	01117
TNC	01136
EH	01143
TOK	01150
COOLIT	01154
COOL2	01162
SCORE	01170
NEXB	01173
DONE	01204
LTAR	01210
HS1	01250
HS2	01251
HS3	01252
HS4	01253
HS5	01254
PRINT.	01255
PRINGO	01261
PROUT	01307
CHAR.	01311
TESP.	01323
TFS1	01337
TYPE.	01341
TAR.	01352
CRLF.	01365
PROCT.	01374
PROCS.	01401
OP1	01406
OPGO	01412
OPS	01420
OPT	01427
HEAD	01436
FM1	01456
FM2	01474
PLIM	01516
ASTFM	01517
AST3	01520
RANKS	01521

ERAP	01522
FLAC	01523
HOP	01524
HSIG	01525
JTFM	01526
OCCT	01527
OCTEM	01530
SAC	01531
TARCT	01532
XBITS	01533
PRINT	101255
CHAR	101311
TESP	101323
TYPE	101341
TAR	101352
CRLF	101365
PROCT	101374
PROCS	101401
CLSF	700001
CLQF	700004
CLON	700044
RSF	700101
RCF	700102
RSA	700104
RRB	700112
RSB	700144
PSF	700201
PCF	700202
PSA	700204
PSB	700244
KSF	700301
KRB	700312
TSF	700401
TCF	700402
TLS	700406
EEM	707702
LEM	707704