

IDENTIFICATION

PRODUCT CODE: MAINDEC-15-D1GA-D (D)
PRODUCT NAME: PDP-15 MEMORY ADDRESS TIMING TEST
DATE CREATED: JANUARY 10, 1970
MAINTAINER: DIAGNOSTICS GROUP
AUTHOR: J. M. GRAETZ/J. RICHARDSON

22

1. ABSTRACT

This program is designed to test the action of long carry chains under the extremely tight timing conditions that obtain at the far end of extended memory. In each register from 10000 to the upper limit of memory, a word of all 1s is stored, then incremented using ISZ. The result is tested to see whether the carry chain was successfully propagated -- the correct result in each case is 000000. Provisions are made for error printout and halt, scoping, and repetition.

2. REQUIREMENTS

2.1 Equipment

A PDP-15 with at least 8192 words of memory.

2.2 Storage

The program occupies registers 00000-00644.
The test field extends from 10000 to the top of memory.

2.3 Preliminary Programs

None

3. LOADING PROCEDURE

The program tape is in absolute binary.
Place the program tape in the reader.
Set the Address Switches to 17700.
Place the BANK MODE Switch on a 1.
Press I/O RESET, then READIN.

4. STARTING PROCEDURE

4.1 Control Switch Settings

| | | |
|------|----|--|
| SW 0 | 1: | Scope Mode. Repeat the current test cycle. |
| | 0: | Normal mode. Do not repeat cycles. |
| SW 1 | 1: | Do not halt on error. |
| | 0: | Normal mode. Halt on occurrence of an error, after printout, if any. |

- SW 2 1: Do not print error data.
 0: Normal mode. Print error data.
- SW 15-17: Number of additional 4K memory banks provided. (Must be at least 1 to execute the test.)

NOTE

SW 0 has precedence over SW 1; that is, if SW 0 is set to 1, SW 1 has no effect. Thus, the error halt is automatically suppressed while the program is operating in scope mode.

4.2 Operator Action

- a. Set the ADDRESS switches to 0200.
- b. Set the control switches as desired (see Section 4.1).
- c. Press Reset, then START.

5. OPERATING PROCEDURE

5.1 Switch Settings

See Section 4.1.

5.2 Operator Action

Normally, the program runs without intervention, until the operator stops it manually. When errors occur, there are several ways to proceed. Here are some common ones.

Set the switches as follows:

$$\begin{array}{ccc} \frac{0}{0} & \frac{1}{0} & \frac{2}{0} & \frac{15}{0} & \frac{16}{0} & \frac{17}{0} \\ & & & \text{number of additional 4K banks} \\ & & & \text{(beyond the standard 4K)} \end{array}$$

Start the program. At the occurrence of the first error, the data will be printed and the program will halt.

- a. Option 1 - To gather more error data, set SW 1 to 1, and press CONT. At the occurrence of each error, the data are printed, and the program continues. When enough information is printed, set SW 0 to 0 and the program will halt after the next printout.

b. Option 2 - For scoping, set SW 0 and SW 2 to 1, connect the scope, and press CONT (SW 1 now has no effect). As long as SW 0 is set to 1, the program will repeat the current test cycle (that is, using the same memory location), regardless of whether the error occurs or not. In this way, an intermittent failure can easily be detected on the scope.

c. Option 3 - If, while scoping, you wish to examine the same signal at the next error site, strike any signal-generating key on the KSR33 keyboard. This will cause the program to break out of the repeat loop and continue the sequential test until the next error occurs, at which point the scope loop resumes. In this way, the operator can step from error to error.

6. ERRORS

6.1 Error Halts and Description

6.1.1 Not Enough Memory - If the operator forgets to set SW 15-17, the program will print

NOT ENOUGH MEMORY
and halt (PC=00047).

6.1.2 Timing Errors - At the first occurrence of an error, the following heading is printed:

EXTENDED MEMORY ISZ TEST
LOCATION CONTENTS

The data are printed in their respective columns. For each error, the LOCATION of the register in error is printed, followed by the actual CONTENTS of the register. In every case, the correct contents should, of course, be 000000. If SW 1 is set to 0, the program halts after the printout (PC=00263).

6.1.3 Skip Failure - If the ISZ instruction fails to skip after incrementing the addressed register to 000000, the following message is printed:

SKIP FAILED

If the data are also in error, these will be printed before the program halts (PC=00263).

6.1.4 Extraneous Errors - If a program interrupt occurs from any device other than the KSR33 keyboard, the program halts at PC=00030, with the I/O status displayed in the AC.

6.2 Error Recovery

- Place the number of additional 4K banks in SW 15-17, and press CONT.
- Press CONT. See Section 5.2 for various operating options.
- Disable the external device causing the interrupt, then press CONT.

6.3 Summary

| <u>C(PC)</u> | <u>C(AC)</u> | <u>Message or Data Printed</u> |
|--------------|--------------|--|
| 00030 | I/O status | None A device other than the KSR33 keyboard caused a program interrupt. Recovery: disable the offending device, and press CONT. |
| 00047 | 000000 | NOT ENOUGH MEMORY Switches 15-17 are all set to 0. Recovery: Set these switches to the number of additional 4K memory banks attached, and press CONT. |
| 00263 | 777777 | Location and contents of register. The contents of the addressed register were not successfully incremented to 000000. Bits appearing as 1 indicate failure of the carry chain. Recovery: Set the control switches as desired, and press CONT. |
| 00263 | 777777 | SKIP FAILED The ISZ instruction failed to skip when the addressed register was incremented to 000000. If the register data were also printed, the skip probably functioned properly; if no data were printed (indicating that the incrementing was successful), then the fault is in the skip itself. Recovery: Set the control switches as desired and press CONT. |

7. RESTRICTIONS

The program will not operate in a machine with less than 8192 words of memory.

8. MISCELLANEOUS

8.1 Execution Time

0.125 x n second for one complete pass through (4n)K of memory.

```

        .TITLE EMTI
        .ABS
        /
        /COPYRIGHT 1970. DIGITAL EQUIPMENT CORP.,
        /MAYNARD, MASS.
        /EXTENDED MEMORY ADDRESS TIMING TEST USING ISZ
        /
00001      .LOC 1
00001      600001      JMP      .
00002      777777      LAW      -1
00003      777777      LAW      -1
00004      777777      LAW      -1
        /
00010      .LOC 10      /AUTOINDEX REGISTERS
00010      000000      AXP      0      /TEXT STRING
00011      000000      AXT      0      /TAB COUNT
00012      000000      AX3      0      /SPARE
        /
00021      .LOC 21      /CAL CATCHER
00021      740040      HLT
00022      620020      JMP*      20
        /
        /KEYBOARD HANDLER
        /
00023      750001      KHAN      CLC
00024      040554      DAC      #KSIG      /SET KEYBOARD SIGNAL
00025      600030      JMP      SPEX
        /
        /SPURIOUS INTERRUPT
        /
00026      700314      SPIN      IORS
00027      740040      HLT      /AC CONTAINS I/O STATUS
00030      200560      SPEX      LAC      SAC
00031      703302      CAF
00032      700042      ION
00033      703344      DBR
00034      620000      JMP*      0
        .EJECT

```

```

/
/PROGRAM STARTS HERE
/
      .LOC 200
00200      703302      IBGIN CAF
00201      200563      LAC          (DAC #SAC
00202      040001      DAC          1
00203      200564      LAC          (KSF
00204      040002      DAC          2
00205      200565      LAC          (JMP SPIN
00206      040003      DAC          3
00207      200566      LAC          (JMP KHAN
00210      040004      DAC          4
00211      750004      LAS
00212      500567      AND          (7          /SWITCHES 15-17
00213      740200      SZA          /EXTENSION ATTACHED?
00214      600221      JMP          INIT          /YES.
00215      200321      LAC          NOMAD          /NO.
00216      100324      PRINT
00217      750040      ENOM HLT:CLA          /GO TELL THE MAN
00220      600200      JMP          IBGIN
/
      .EJECT
```

```

/INITIALIZING SEQUENCE
/
00221 700042 INIT ION
00222 740001 CMA
00223 040552 DAC #BANKS /BANK COUNT
00224 750000 CLA /FORM LIMITING ADDRESS
00225 300570 ADD (10000
00226 440552 ISZ BANKS
00227 600225 JMP .-2
00230 040555 DAC #LIMIT /ADDRESS OF HIGHEST REGISTER
00231 750001 CLC
00232 040553 DAC #HSIG /SET HEADING SIGNAL
00233 040554 DAC KSIG /PRESET KEYBOARD SIGNAL
00234 200570 INIT2 LAC (10000 /SET Y-ADDRESS POINTER
00235 040562 DAC #YAP
/
/MAIN TEST SEQUENCE
/
00236 707702 TEST EEM
00237 750001 CLC
00240 060562 DAC* YAP /ALL 1S TO Y
00241 460562 ISZ* YAP /THIS IS IT!
00242 600312 JMP SKIF /SKIP FAILED
00243 220562 TEY LAC* YAP /GET C(Y)
00244 741200 SNA /C(Y)=0?
00245 600306 JMP TOK /YES; TEST OK
00246 140554 DZM KSIG /NO: CLEAR KRD SIGNAL
00247 707704 LEM
00250 750004 LAS
00251 742010 RTL /SW-2
00252 751100 SPA!CLA /PRINTING?
00253 600267 JMP TNG /NO.
00254 540553 SAD HSIG /YES. PRINT HEADING?
00255 600261 JMP TELL /NO.
00256 140553 DZM HSIG /YES. CLEAR SIGNAL
00257 340322 TAD HEAD
00260 100324 PRINT
/
00261 200562 TELL LAC YAP /GET ADDRESS
00262 100444 PROCS /PRINT IT
00263 100415 TAB
00264 220562 LAC* YAP
00265 100437 PROCT /PRINT CONTENTS
00266 100430 CRLF /NEXT LINE
/
.EJECT

```



```

                                /SWITCH TESTERS
                                /
00267  750004  TNG  LAS                                /HERE WHEN THERE'S AN ERROR
00270  740010          RAL                                /SW0 AND SW1
00271  740500          SNL!SMA                            /SCOPING OR HALT?
00272  750041  E1   HLT!CLC                            /HALT
00273  750004          LAS
00274  750100          SMA!CLA                            /SCOPING?
00275  600301          JMP      XPO                        /NO.
00276  200554          LAC      KSIG                       /YES
00277  741200          SNA
00300  600236          JMP      TEST                       /IS KBD SIGNAL SET?
                                                    /NO. REPEAT THE CYCLE

00301  440562  XPO  ISZ      YAP                            /YES. GO ON
00302  200555          LAC      LIMIT
00303  540562          SAD      YAP                            /END OF MEMORY?
00304  600234          JMP      INIT2                       /YES. START AGAIN
00305  600236          JMP      TEST                       /NO. NEXT CYCLE

00306  750004  TOK  LAS                                /HERE WHEN NO ERROR
00307  751100          SPA!CLA                            /SCOPING?
00310  600236          JMP      TEST                       /YES
00311  600301          JMP      XPO                        /NO.

                                /
                                /SKIP FAILED
                                /
00312  750004  SKIF LAS
00313  742010          RTL
00314  741100          SPA
00315  600243          JMP      TEY
00316  200323          LAC      SKAD                            /SW-2
00317  100324          PRINT
00320  600243          JMP      TEY                            /PRINTING?
                                                    /NO.
                                                    /YES

00321  000500  NOMAD NOMEM-1
00322  000520  HEAD  HEADER-1
00323  000512  SKAD  SKIM-1

                                /
                                .EJECT

```

```

                                /TEXT PRINTER
                                /
00324 000000 PRINT. 0
00325 040010 DAC AXP /ADDRESS OF TEXT STRING
00326 220010 PRINGO LAC* AXP /FIRST PAIR-WORD
00327 040550 DAC #ASTEM
00330 100354 CHAR /POSITION FIRST CHARACTER ...
00331 100366 TESP /...SECOND...
00332 100354 CHAR /...PUT THE THIRD TOGETHER...
00333 100366 TESP
00334 100354 CHAR
00335 500571 AND (170
00336 040551 DAC #AST3
00337 220010 LAC* AXP /SECOND PAIR-WORD
00340 744010 RCL; RTL /ADJUST
00341 742010
00342 040550 DAC ASTEM
00343 740010 RAL /3RD CHAR LOW-ORDER BITS
00344 500567 AND (7
00345 240551 XOR AST3 /ASSEMBLE THE PARTS
00346 100366 TESP
00347 100354 CHAR /...FOURTH...
00350 100366 TESP
00351 100354 CHAR /...AND FIFTH
00352 100366 TESP
00353 600326 JMP PRINGO
                                /
                                .EJECT

```

```

/TEXT PRINTER SUBROUTINES
/
/1. POSITION CHARACTER
/
CHAR. 0
00354 000000 LAC      ASTEM
00355 200550 RCL
00356 744010 RTL;    RTL;    RTL
00357 742010
00360 742010
00361 742010
00362 040550 DAC      ASTEM      /NEXT CHAR. NOW LEFT-JUSTIFIED
00363 740010 RAL      /LAST BIT IN PLACE
00364 500572 AND      (177      /ISOLATE 7 BITS
00365 620354 JMP*    CHAR.
/
/2. TEST AND PRINT CHARACTER
/
TESP. 0
00366 000000 SNA
00367 741200 JMP*    TESP.      /0 CHARACTER? (ASCII FILLER)
00370 620366 SAD      (177      /YES. IGNORE IT.
00371 540572 JMP*    PRINT.     /TERMINATOR?
00372 620324 XOR      (200      /YES. GO AWAY
00373 240573 SAD      (211      /NO. FINISH THE ASCII-CODE
00374 540574 JMP      TETAB     /TAB?
00375 600402 TYPE     /YES.
00376 100404 SAD      (212      /NO.
00377 540575 DZM      TABCT     /LF?
00400 140561 JMP*    TESP.     /YES. CLEAR TAB
00401 620366 TETAB  TAB
00402 100415 JMP*    TESP.
00403 620366
/
/3. TYPE A CHARACTER
/
TYPE. 0
00404 000000 IOF
00405 700002 TLS
00406 700406 TSF
00407 700401 JMP      .-1
00410 600407 TCF
00411 700402 ION
00412 700042 ISz     TARCT
00413 440561 JMP*    TYPE.
00414 620404
/
.EJECT

```

```

/4. TABULATOR
/
00415 000000 TAB. 0
00416 200561 LAC #TABCT /COUNT OF SPACES
00417 340576 TAD (-12 /REDUCE MODULO 10
00420 740100 SMA /NEGATIVE YET?
00421 600417 JMP .-2 /NO.
00422 040011 DAC AXT /YES, STORE SPACE COUNT
00423 200577 LAC (240 /(SPACE)
00424 100404 TYPE
00425 440011 ISZ AXT /DONE?
00426 600424 JMP .-2 /NO.
00427 620415 JMP* TAB. /YES

```

/5. CARRIAGE RETURN LINE FEED

```

/
00430 000000 CRLF. 0
00431 200600 LAC (215
00432 100404 TYPE
00433 200575 LAC (212
00434 100404 TYPE
00435 140561 DZM TABCT
00436 620430 JMP* CRLF.

```

```

/
100324 PRINT=JMS PRINT.
100354 CHAR=JMS CHAR.
100366 TESP=JMS TESP.
100404 TYPE=JMS TYPE.
100415 TAB=JMS TAB.
100430 CRLF=JMS CRLF.

```

/ .EJECT

```

/OCTAL PRINT SUBROUTINE
/
/ENTRY TO PRINT LEADING ZEROS
/
00437 000000 PROCT. 0
00440 040557 DAC #OCTEM
00441 200466 LAC OPS+3 /NOP
00442 100451 JMS OP1
00443 620437 JMP* PROCT.

/
/ENTRY TO REPLACE LEADING ZEROS WITH SPACES
/
00444 000000 PROCS. 0
00445 040557 DAC OCTEM
00446 200601 LAC (SZA
00447 100451 JMS OP1
00450 620444 JMP* PROCS.

/
/PRINTER
/
00451 000000 OP1 0
00452 040463 DAC OPS /SET SWITCH
00453 777772 LAW -6
00454 040556 DAC #OCCT /DIGIT COUNTER
00455 200557 OPGO LAC OCTEM
00456 744010 RCL; RTL
00457 742010
00460 040557 DAC #OCTEM
00461 740010 RAL
00462 500567 AND (7
00463 740040 OPS XX /SZA OR NOP
00464 600472 JMP OPT
00465 440556 ISZ OCCT /LEADING ZERO, COUNT DIGITS
00466 740000 NOP
00467 200577 LAC (240 /TYPE A SPACE
00470 100404 TYPE
00471 600455 JMP OPGO

/
00472 240602 OPT XOR (260 /ASCII MAKER
00473 100404 TYPE
00474 200466 LAC OPS+3 /REPLACE SZA WITH NOP
00475 040463 DAC OPS
00476 440556 ISZ OCCT
00477 600455 JMP OPGO
00500 620451 JMP* OP1

/
100444 PROCS=JMS PROCS.
100437 PROCT=JMS PROCT.
/
.EJECT

```

/MESSAGES

| | | | | |
|-------|--------|--------|--------|---|
| 00501 | 064251 | NOMEM | .ASCII | <15><12>'NOT ENOUGH MEMORY'<15><12><177> |
| 00502 | 647650 | | | |
| 00503 | 202131 | | | |
| 00504 | 647652 | | | |
| 00505 | 436204 | | | |
| 00506 | 046612 | | | |
| 00507 | 466372 | | | |
| 00510 | 254432 | | | |
| 00511 | 053760 | | | |
| 00512 | 000000 | | | |
| 00513 | 516271 | SKIM | .ASCII | 'SKIP FAILED'<15><12><177> |
| 00514 | 150100 | | | |
| 00515 | 432031 | | | |
| 00516 | 146212 | | | |
| 00517 | 420321 | | | |
| 00520 | 277400 | | | |
| 00521 | 064250 | HEADER | .ASCII | <15><12>'EXTENDED MEMORY ISZ TEST.'<15><12><12> |
| 00522 | 554250 | | | |
| 00523 | 426350 | | | |
| 00524 | 442610 | | | |
| 00525 | 202330 | | | |
| 00526 | 546636 | | | |
| 00527 | 512624 | | | |
| 00530 | 044646 | | | |
| 00531 | 551012 | | | |
| 00532 | 442646 | | | |
| 00533 | 521341 | | | |
| 00534 | 505024 | | | |
| 00535 | 462370 | | .ASCII | 'LOCATION'<11>'CONTENTS'<15><12><12><177> |
| 00536 | 340650 | | | |
| 00537 | 446371 | | | |
| 00540 | 604606 | | | |
| 00541 | 476352 | | | |
| 00542 | 442634 | | | |
| 00543 | 522461 | | | |
| 00544 | 505024 | | | |
| 00545 | 774000 | | | |
| 00546 | 000000 | | | |
| 00547 | 000603 | | .SIZE | |
| | 000000 | | .END | |
| | 00563 | 040560 | *L | |
| | 00564 | 700301 | *L | |
| | 00565 | 600026 | *L | |
| | 00566 | 600023 | *L | |
| | 00567 | 000007 | *L | |
| | 00570 | 010000 | *L | |
| | 00571 | 000170 | *L | |
| | 00572 | 000177 | *L | |
| | 00573 | 000200 | *L | |
| | 00574 | 000211 | *L | |

PAGE 10

EMTI

EMTI

| | | |
|-------|--------|----|
| 00575 | 000212 | *L |
| 00576 | 777766 | *L |
| 00577 | 000240 | *L |
| 00600 | 000215 | *L |
| 00601 | 740200 | *L |
| 00602 | 000260 | *L |

SIZE=00603

NO ERROR LINES

ASTEM 00550
AST3 00551
AXP 00010
AXT 00011
AX3 00012
BANKS 00552
CHAR 100354
CHAR. 00354
CLOF 700004
CLON 700044
CLSF 700001
CRLF 100430
CRLF. 00430
EEM 707702
FNOM 00217
F1 00272
HEAD 00322
HEADER 00521
HSIG 00553
IBGIN 00200
INIT 00221
INIT2 00234
KHAN 00023
KRB 700312
KSF 700301
KSIK 00554
LEM 707704
LIMIT 00555
NOMAD 00321
NOMEM 00501
OCCT 00556
OCTEM 00557
OPGO 00455
OPS 00463
OPT 00472
OP1 00451
PCF 700202
PRINGO 00326
PRINT 100324
PRINT. 00324
PROCS 100444
PROCS. 00444
PROCT 100437
PROCT. 00437
PSA 700204
PSR 700244
PSF 700201
RCF 700102
RRR 700112
RSA 700104
RSB 700144
RSF 700101
SAC 00560
SKAD 00323
SKIF 00312

PAGE 12 EMTI EMTI

| | |
|-------|--------|
| SKIM | 00513 |
| SPEX | 00030 |
| SPIN | 00026 |
| TAR | 100415 |
| TARCT | 00561 |
| TAR. | 00415 |
| TCF | 700402 |
| TELL | 00261 |
| TESP | 100366 |
| TESP. | 00366 |
| TEST | 00236 |
| TETAB | 00402 |
| TEY | 00243 |
| TLS | 700406 |
| TNG | 00267 |
| TOK | 00306 |
| TSF | 700401 |
| TYPE | 100404 |
| TYPE. | 00404 |
| XPO | 00301 |
| YAP | 00562 |
| .EOT | 00000 |

| | |
|--------|--------|
| .EOT | 00000 |
| AXP | 00010 |
| AXT | 00011 |
| AX3 | 00012 |
| KHAN | 00023 |
| SPIN | 00026 |
| SPEX | 00030 |
| IBGIN | 00200 |
| ENOM | 00217 |
| INIT | 00221 |
| INIT2 | 00234 |
| TEST | 00236 |
| TEY | 00243 |
| TELL | 00261 |
| TNG | 00267 |
| E1 | 00272 |
| XPO | 00301 |
| TOK | 00306 |
| SKIF | 00312 |
| NOMAD | 00321 |
| HEAD | 00322 |
| SKAD | 00323 |
| PRINT. | 00324 |
| PRINGO | 00326 |
| CHAR. | 00354 |
| TESP. | 00366 |
| TETAB | 00402 |
| TYPE. | 00404 |
| TAB. | 00415 |
| CRLF. | 00430 |
| PROCT. | 00437 |
| PROCS. | 00444 |
| OP1 | 00451 |
| OPGO | 00455 |
| OPS | 00463 |
| OPT | 00472 |
| NOMEM | 00501 |
| SKIM | 00513 |
| HEADER | 00521 |
| ASTEM | 00550 |
| AST3 | 00551 |
| BANKS | 00552 |
| HSIG | 00553 |
| KSIG | 00554 |
| LIMIT | 00555 |
| OCCT | 00556 |
| OCTEM | 00557 |
| SAC | 00560 |
| TARCT | 00561 |
| YAP | 00562 |
| PRINT | 100324 |
| CHAR | 100354 |
| TESP | 100366 |
| TYPE | 100404 |
| TAR | 100415 |

| | |
|-------|--------|
| CRLF | 100430 |
| PROCT | 100437 |
| PROCS | 100444 |
| CLSF | 700001 |
| CLOF | 700004 |
| CLON | 700044 |
| RSF | 700101 |
| RCF | 700102 |
| RSA | 700104 |
| RRB | 700112 |
| RSR | 700144 |
| PSF | 700201 |
| PCF | 700202 |
| PSA | 700204 |
| PSB | 700244 |
| KSF | 700301 |
| KRB | 700312 |
| TSF | 700401 |
| TCF | 700402 |
| TLS | 700406 |
| EEM | 707702 |
| LEM | 707704 |