PROJECT MAC

Computer Systems Research Division

October 9, 1973

Request for Comments No. 39

A CPU SPEED MEASUREMENT TOOL

by J. H. Saltzer

The enclosed writeup, in SPS format, describes a software tool I have found very useful in making measurements of the 6180 processor. It may be found in my directory, under the name

">udd>CompSys>Saltzer>mip_test".

---

mip_test

tool

April 30, 1973

Purpose

The mip_test command is a diagnostic tool used to measure the performance of the central processor and identify sources of interference with its normal running speed. The command performs a standard instruction sequence 1000 times, noting the time required for each of the 1000 runs. It then prints a table of run times (in microseconds), run speeds (in millions of instructions per second) and number of observations. Excessively long run times (more than about 3 times normal) are assumed to be the result of interrupts, and are summarized, rather than included in the table. The primary use of this command is to verify that the processor is working correctly.

Usage

mip_test -sequence_option- -sleep_count- -margin-

where:

1) sequence_option — if present indicates which of four instruction sequences is to be used. The following are recognized:

ada	ada instructions
epp	epp instructions with ordinary address
eppi	eppi instructions with indirect address
spr	spr instructions
mix	"Multics mix" of instructions

If sequence_option is omitted, "mix" is used.

2) sleep_count — if present, is a decimal integer indicating the number of times the experiment should be repeated, with a ten-second pause between repeats. A single summary table is printed, combining the repeats. If sleep_count is omitted, a value of 1 is used. A sleep_count larger than 1 may be useful in making the program run in different memory boxes at different times.

3) margin — if present, indicates that shorter or longer instruction sequence than usual should be used. The value margin = "short" produces an instruction sequence about 50% shorter, while the value margin = "long" produces an instruction sequence about 50% longer. The longer and shorter instruction sequences are sometimes useful in interpreting abnormal results.

Notes

The actual run timing is performed in a machine language subroutine which reads the calendar clock, executes the test sequence, and reads the calendar clock again. To control the effect of the time required for the clock reading itself, a test sequence requiring 200-300 microseconds is used.

The machine language program is an impure procedure, so that all instruction and operand references for a single test sequence can be concentrated in a single page and also located in the same memory controller. (If interlace is used, the test sequence may be spread among memory controllers despite its concentration in a single page.) For this reason, the test may run at normal speed even if part of the processor associative memory is disabled -- only one or two AM registers are needed for normal-speed operation.

Because it is an impure procedure, some care is required to run it. The internal alm procedure is named "mipt", and is placed in the Multics storage system with the copy switch set on. This switch guarantees that when the program is used, a private copy is first made for the user, in his process directory, using a standard unique identifier for its segment name.

If there is no interference, all 100 test sequences should run with the same speed. In practice, one usually observes about 70-90% of the sequences to be at one speed, and the remaining 10-30% to be at slower speeds, corresponding to various combinations of memory interference caused by I/O or another processor. In addition, the different CPU's and memory boxes have different speeds in a range of about $\pm$ 1%; some experiments will exhibit these differences.

Program mip_test calls the command pcd to get a listing of cpu's and memories, both before and after the experiment. Use of the pcd command requires system programming (phcs_) privileges.

The thirty instruction sequence used as a "Multics mix" is as follows:

```
eppbp     its       "1
spr ibp   its       "2
ldaq      bp|3      "3
ada       bp|4      "4
sta       w         "5
ldaq      y         "6
mpy       1,dl      "7
staq      w         "8
spribp    its       "9
tra       1,ic      "10
lda       bp|0,*    "11
anaq      bp|0      "12
qrl       34        "13
sta       w         "14
aos       x         "15
eppbp     its       "16
lda       w         "17
ldaq      x         "18
eraq      z         "19
cmpq      =0        "20
tnz       1,ic      "21
lls       34        "22
spr ibp   its       "23
fld       0,dl      "24
ada       y         "25
sba       z         "26
sta       w         "27
lda       0,du      "28
eax1      -1,1      "29
tnz       -29,ic    "30
```

The data area addressed by this sequence is in the same page of memory as the instruction sequence, and is declared as follows: (location "its" is initialized with a pointer to itself.)

```
          even
          bss       clk1,2
its:      dec       0
          dec       0
x:        dec       431
y:        dec       795
z:        dec       87531
w:        dec       0
```

On the following three pages are sample outputs of the mip_test command, on the HISI 6180 at M.I.T. The occasional fast execution of the "spr" instruction are unexplained.

```
mip_test mix

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

10 trials of over 1000 microseconds.
musecs mips
   282  .644 0
   280  .648 1
   279  .652 51
   277  .656 948
   275  .660 0
average mips =  .659
each trial 182 instructions

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

r 1703  1.055  3.450  107

mip_test mix long 10

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

73 trials of over 1150 microseconds.
musecs mips
   430  .632 0
   427  .636 1
   425  .640 35
   422  .644 57
   419  .648 2790
   417  .652 118
   414  .656 92
   412  .660 6907
   409  .664 0
average mips =  .658
each trial 272 instructions

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

r 1706  7.825  7.358  267
```

```
mip_test spr

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

16 trials of over 1000 microseconds.
musecs mips
   409  .444 0
   406  .448 112
   402  .452 873
   399  .456 0
   395  .460 0
   392  .464 0
   388  .468 0
   385  .472 0
   382  .476 0
   379  .480 0
   376  .484 0
   372  .488 0
   369  .492 0
   366  .496 0
   364  .500 0
   361  .504 0
   358  .508 15
   355  .512 0
average mips =  .453
each trial 182 instructions

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

r 1709  1.380  1.890  61

mip_test ada

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

3 trials of over 1000 microseconds.
musecs mips
   203 1.188 0
   203 1.192 461
   202 1.196 539
   201 1.200 0
average mips = 1.195
each trial 242 instructions

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

r 1709  .943  1.572  40
```

```
mip_test epp

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

5 trials of over 1000 microseconds.
musecs mips
   280  .648 0
   279  .652 1
   277  .656 754
   275  .660 245
   274  .664 0
average mips =  .660
each trial 182 instructions

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

r 1656  1.208  4.658  111

mip_test eppi

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

4 trials of over 1000 microseconds.
musecs mips
   363  .336 0
   358  .340 1000
   354  .344 0
average mips =  .341
each trial 122 instructions

cpu  b  6
mem  b  128.  on
mem  c  128.  on
mem  a  128.  off

r 1657  1.208  4.558  84

hmu

Multics 20.12bx, load 22.0/50.0; 22 users
Absentee users 0/2

r 1657  .110  .136  11

Today is October 8, 1973  @
```