To: Distribution

From: J. H. Saltzer

Date: March 18, 1975

Subject: Initial measurements of cache speed

Between January 23 and January 27, 1975, there was an opportunity to compare the relative raw speeds of M.I.T. 6180 processor B with the newer 68/80 processor C. Processor C contains a cache memory intended to make most memory references faster. During this period, CPU C was run both with the cache operating and with the cache switched off. In addition, we have available a calibration run from April, 1973, before certain speedup changes were (apparently) added to CPU B. Finally, at the end of that period, 6180 CPU A was reinstalled, after it was retrofitted with a cache memory. Thus a wide variety of interesting comparisons can be made.

Speed measurements were made as follows, using an ALM impure procedure which fit into one page and contained both the instructions and all data operands:

a. read calendar clock
b. enter a 30 instruction loop and go around it n times (n = 3 to 9)
c. read calendar clock
d. note difference in clock readings in a histogram
e. repeat a-d 1000 times without pause
f. repeat a-e 20 times with 10 second pause
g. display the histogram of clock reading differences.

In the resulting display, the mode of the histogram (usually containing 80-90% of the trials) was taken to represent the normal speed of the processor. The above experiment was performed twice, once with n = 3 and once with n = 9, and the difference between the two times was taken to be the time required to execute the different number of instructions. (Thus cancelling any measurement error due to the non-zero time required to read the calendar clock and start and end the loops.)

The experiment was performed with four "pure" sequences of identical instructions (ADA, SPR, EPP, and EPP with indirect address) and with a mixed sequence of 30 (non-EIS) instructions chosen to simulate typical Multics PL/I-generated code. Table I displays the result.

1 Instruction Sequence	2 CPU A 4/73	3 CPU B 1/75	4 CPU C cache on 1/75	5 raw performance gain B → C	6 CPU C cache off 1/75	7 CPU A cache off 1/75	8 estimated CPU A with 75% hit ratio	9 estimated performance gain with 75% hit ratio
ADA	810 ns	830 ns	690 ns	1.20	1050 ns	1000 ns	770 ns	1.08
EPP	1490 ns	1490 ns	910 ns	1.63	1140 ns	1080 ns	950 ns	1.56
EPP,*	2940 ns	2940 ns	1730 ns	1.70	2600 ns	2460 ns	1910 ns	1.54
SPR	2230 ns	1360 ns	1260 ns	1.08	2250 ns	2180 ns	1490 ns	.91
Multics Mix	.66 mips	.71 mips	.92 mips	1.31	.65 mips	.71 mips	.87 mips	1.23

Table I CPU performance measurements with and without cache. Instruction times are in nanoseconds (average for 100 or more identical, consecutive instructions). "Mix" measurements are in millions of instructions per second. All measurements have a precision (repeatability) of about ± 1%.

The measurements of Table I should be interpreted with the reservation that it is assumed that the cache design permits all of the test program and its data to fit into the cache simultaneously, without self-interference resulting from different regions of the program accidentally mapping into the same cache locations. If this assumption is wrong, CPU C with cache on is faster than shown by these measurements.

We make several interesting observations from Table I:

1. Columns 2 and 3, which report 6180 measurements in 1973 and in 1975, are essentially the same, except for the time of the SPR instruction. Apparently, sometime between 4/73 and 1/75 a modification was made to CPU B to allow address preparation look ahead (previously disabled) on SPR instructions, producing a 6% average performance gain on the Multics mix. Note that the apparently perfectly identical timings of the EPP instructions on two different CPU's two years apart merely means that the original raw data indicated that the same integral number of microseconds were measured for the same 140-instruction sequence.

2. The raw speed of CPU C with cache on, shown in column 4, ranges from 8% to 70% faster than CPU B, with the Multics mix running 30% faster, as indicated in column 5. Measurements of CPU A with cache on were identical to within the precision of measurement (± 1%).

3. CPU C with its cache off is significantly slower than CPU B, for ADA and SPR instructions. Performance of the EPP instruction is much better than before. Note that CPU C apparently disables address preparation overlap on control unit store instructions, so that with the cache off, the SPR instruction has a timing comparable to the 1973 time of CPU A.

4. CPU C is said to be attached with CPU-memory cables about 20 feet longer than normal, a temporary arrangement because of the processor swap out strategy. This longer cable adds about 45 ns to the time required for a memory reference. An ADA instruction requires one operand fetch, and each pair of ADA instructions requires an instruction fetch. Thus an ADA instruction should encounter an average of 67.5 ns of extra delay, although some of that may happen to be overlapped with instruction execution. This consideration applies only to the measurements of CPU C with the cache off. Column 7 provides the corresponding figures for 1975 CPU A with cache off. The effect of the cable length is indeed about what was expected.

5. Comparison of columns 3 and 7 suggests that if CPU A is run with the cache switched off, a negligible performance change relative to not having a cache at all is to be expected. This result is very mix-sensitive, however, since the SPR instruction is 38% slower while the EPP instruction is 37% faster.

6. Column 4 represents the timing of a cache CPU for a 100% cache hit ratio. Column 7 represents the timing of a cache CPU for a 0% cache hit ratio. To a first approximation, hit ratios between 0% and 100% should result in performance linearly interpolated between columns 4 and 7. Column 8 and 9 estimate the timings and performance gain, respectively, for a 75% hit ratio.

To gain a better feel for the effect of hit ratio on performance, figure 1 shows the linear interpolation of the "Multics mix" performance for hit ratios between 0% and 100%.

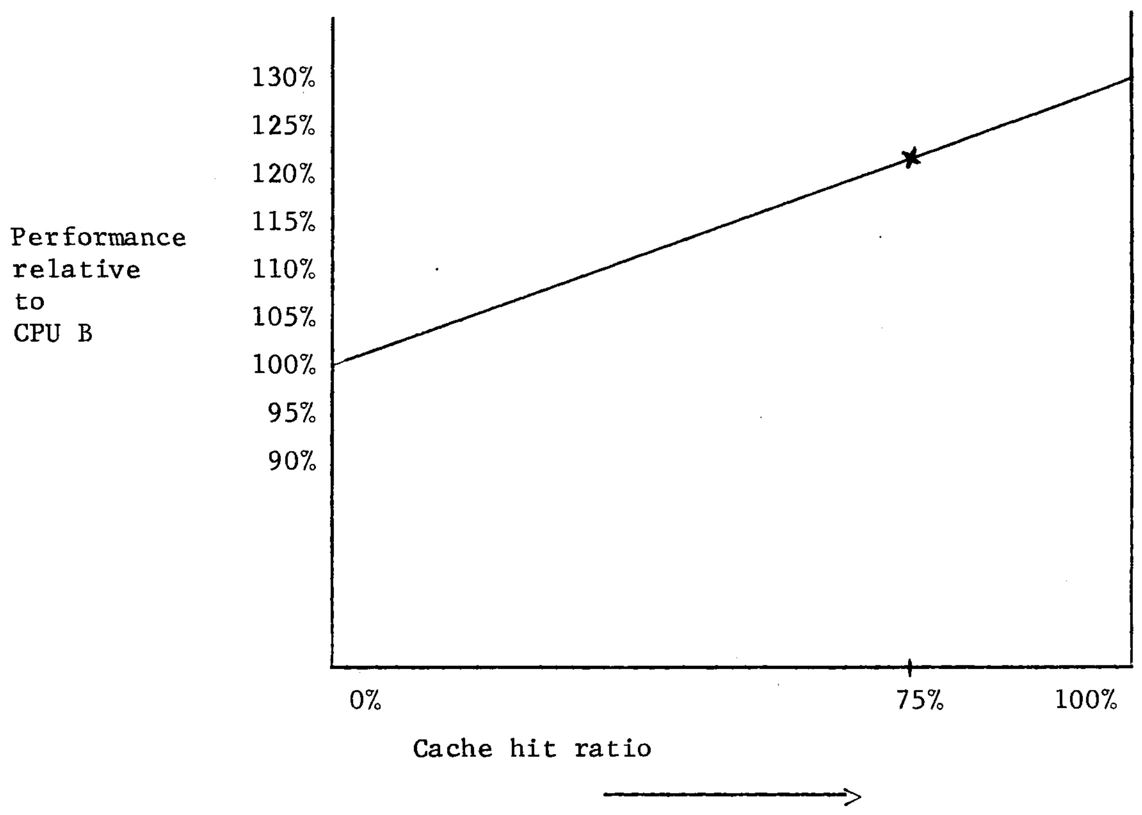

Figure 1 -- CPU A performance relative to CPU B, as a function of cache hit ratio.