Retrospective:

Characterization of Processor Performance in the VAX-11/780

Joel S. Emer

Digital Equipment
Shrewsbury, MA, 01545
Joel.Emer@Digital.com

Douglas W. Clark

Dept. of Computer Science
Princeton University, Princeton, NJ 08544
doug@cs.princeton.edu

In reminiscing with early VAX designers about the work in this paper, it has been difficult to recall how startlingly primitive our performance knowledge and approaches once were. While inside Digital and in the larger architecture research community we are now thoroughly indoctrinated in the quantitative approach to computer architecture and design, of which this paper is an early example, the situation in the early 1980's was quite different. In particular, while the VAX-11/780, which was introduced in 1978, was probably the preeminent timesharing machine on university campuses at that time, very little was known about how it worked or exactly what its performance was.

In particular, before 1980, even inside Digital the fact that some benchmarks ran at less than the widely-believed 1 MIPS was known to only a very small number of people. And the fact that on real multiuser workloads the 11/780 typically executed instructions at only 0.5 MIPS was apparently unknown. Furthermore, somewhat embarrassingly, both facts were unknown to the architects of some of the successor machines. That meant that those designs were optimizing to the presumed 5 average CPI of the 11/780, where in fact another 5 cycles per instruction were totally unaccounted for. It was only following some other measurements by one of us (Joel) in which a frequency counter was hooked up to record MIPS and he was shocked to read 0.5 MHz where he expected 1.0 MHz, that a more widespread account of the 0.5 MIPS rating was propagated. Still, so widely believed was the 1.0 MIPS number, in fact, that one of our ISCA referees didn't believe the data, making the "mandatory" recommendation to "explain why Table 8 and 1st bullet, pg. 23, seem to imply average VAX 780 instruction takes > 2 us; should be ~1 us."

In addition, while we have become accustomed to single chip microprocessors with minimal interfaces to probe their internal operations, the VAX 11/780 CPU spanned about 20 boards. One such board was the microcode store, which directed much of the behavior of the machine. That meant that one could probe the backplane of the machine to determine the address of each microinstruction executed. That's exactly what the measurement tool described in the paper was able to do. Furthermore, a microcoded processor like the 11/780 reveals a huge amount of detailed behavior this way, some of which was reported in the paper included in this volume.

While it would be nice to claim that all the work in the paper was premeditated as a comprehensive characterization of the 11/780, the microPC histogram tool used in the study was actually inspired by a single question that it wouldn't answer. It was probably late in 1980, and the company was in the early stages of the design of the VAX 8200, the first microprocessor VAX. Although it was a microprocessor, it wasn't on one chip, but the CPU core spanned three chips, not including the cache. Furthermore, the microcode had to be on an additional five chips. Since chip crossings were expensive, it was suggested that perhaps a two-level hierarchical microcode store would perform better. Thus, some small number of microinstructions could be included in the processor chip, and the remainder would live in the microcode chip. But with different latencies for different microinstructions, what would the performance be? The answer of course depended on the execution rate of each microinstruction. Unfortunately, we had little idea what the actual rates were.

The way to answer this question was obvious. Measurements of PC histograms for applications were commonplace, so why not measure the microPC histogram? Of course, as is invariably the case for questions that arise during a design, there was no time to conduct an extensive new study, especially one that involved building new measurement hardware. So a decision was made to build a single level control store for the 8200, as much for hardware complexity arguments as performance arguments. But the idea of a device to characterize microcode behavior was established, and given our preexisting belief in the value of accurate performance characterization, it was pursued in the expectation that next time we would be prepared with data to answer many other questions as they arose.

By the Fall of 1981, the first set of measurements was completed. Figure 1 shows the original graph that could answer the design question we were too late to answer. In addition, there was also a wealth of data on many of the arcane facets of use of the VAX architecture and the 11/780 implementation. At Dick Sites' suggestion, we created an annotated microcode listing that showed the relative execution count of each 11/780 microinstruction. Those listings were indispensable to several generations of VAX microcoders, who used them to determine the relative frequencies of different cases or typical microcode loop counts, and to budget microcode space. In addition, it was used to justify a variety of hardware/microcode tradeoffs.

Probably most significant was the two-dimensional instruction class versus operation cycle count table that appears as Table 8 in the paper. While such breakdowns of architectural and implementation statistics seems obvious and essential today, there were many novel applications at the time. This information was used to quell an internal attack on the (ultimately correct) performance claims of the VAX 8800. And the VAX 9000 architect carried around a marked up copy of the diagram with crossed-out entries, and updated values to justify the performance expectations of his machine. We believe that this data used in the early 80's was the most compelling evidence for performance claims that DEC designers had used to date, and was instrumental in establishing a firmly quantitative approach to performance inside the company.

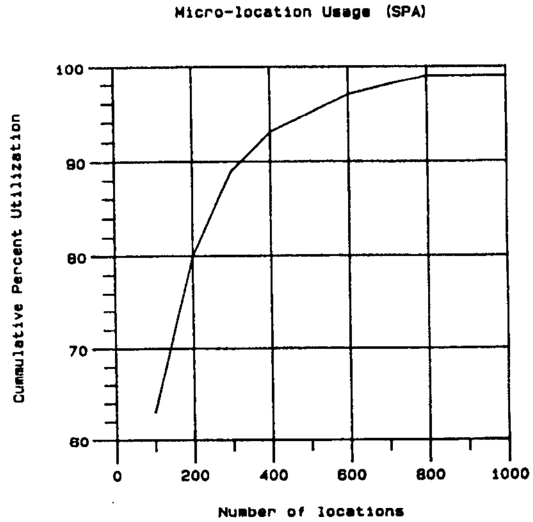
On the other hand, the most fun we had with the data happened in various design meetings, when, as seemed standard practice in those days, some clever designer would claim that a monumental performance gain could be achieved if only some cache were enlarged or the translation buffer were improved. It became a pleasant avocation to short-circuit these discussions with hard data, which inevitably showed that no single clever idea could cut VAX CPI significantly. Of course, this was not universally appreciated, as evidenced by the remark of a very senior designer, who in response to the interjection of a measured fact into a heated discussion, said, "Boy, you ruin all our fun — you have data."

As is often the case with industrial research, there was not a large incentive to publish results immediately, and this particular work was available within DEC for over two years before it was submitted to ISCA. There was also some understandable concern over the sensitivity of the data. In the end, we (and the corporate reviewers) decided that releasing the data was the right thing to do, as long as we didn't make the blunt observations that the 11/780 was only a 0.5 MIPS machine. Thus, noting that it was a little over 10 CPI and had a 200 ns. cycle time was okay, but no MIPs number was to appear.

In the end, we have been pleased by the acceptance of this paper as an example of the quantitative approach to computer architecture. We also have been pleased by the use of some other techniques exhibited by the paper, such as the separation of architectural and implementation statistics, the use of per-instruction metrics, and the use of better benchmark programs, especially those that include multiple users and system activity. It also seems clear to us in retrospect that this paper provided a service to designers of competitive machines (especially in academe), by quantitatively characterizing the most commonly used benchmark processor.

Both authors of this paper have continued to work on computer architecture. Joel has remained at Digital, and has worked on performance evaluation for a number of VAX processors. Doug was a designer of the VAX 8800 family, and worked on two further VAXs that never shipped. Both participated in the small corporate taskforce that led to the creation of the Alpha architecture. At that point their paths diverged, as Doug left Digital for an academic position at Princeton, while Joel has remained at Digital doing architectural research on various Alpha processors.

Figure 1. Original graph of micro-location usage


Micro-location Usage (SPA)
Cummulative Percent Utilization
Number of locations