RETROSPECTIVE:

RISC I: A Reduced Instruction Set Computer

David A. Patterson and Carlo H. Séquin

Computer Science Division
University of California, Berkeley, CA 94720
{pattrsn,sequin}@CS.Berkeley.EDU

This 1981 paper was written as part of the RISC movement that began to flourish in the early 1980s. The three groups leading the charge were at IBM, Berkeley, and Stanford.

IBM was the earliest, focusing on advances in compiler technology and instruction sets that compilers could use to get good performance without the need for a microcode interpreter. Their targets were a 24-bit ECL minicomputer for hardware, called the 801, and a programming language they invented called PL8, and their competition was the IBM 370 family of computers.

As the introduction to this paper suggests, the Berkeley effort was in trying to design an instruction set that made sense for a single VLSI chip. Our group did not include compiler experts, so that was not something that we were pushing. Our targets were a 40,000 transistors, 32-bit NMOS microprocessor and the programming language C and UNIX operating system, and the competition was the VAX-11/780, a relatively new machine that was making big waves in the marketplace.

The Stanford effort was also interested in a 32-bit single chip microprocessor, called MIPS for Microprocessor without Interlocked Pipeline Stages, and since Hennessy knew compilers they pushed it as well. They concentrated on the Pascal language, and while they didn't typically compare to other machines, occasionally they compared to the PDP-10.

The Berkeley RISC effort was inspired in large part by Patterson's reaction to a sabbatical he took at DEC in Fall 1979, and by our goal to make our architecture courses "hands-on" and as relevant as possible. This was the first time a university planned to actually build a complete microprocessor on a chip, and many people let us know that we had almost zero chance of success. So we were well aware that we had to keep the structure and the logic of this chip as simple as we could get away with. Séquin, at that time, was involved as a consultant in the Mead-Conway revolution of getting universities involved in chip design. Having previously built several chips at Bell Labs, he was more aware of what it would take to make a working chip, but tried to hide his anxieties in order not to dampen the enthusiasm for the project.

Patterson had worked on microprogramming tools for his Ph.D., and that was what he had been helping with at DEC. He wondered about building a VAX as a single chip, especially given all the microcode bugs which were often patched in the field. (Indeed, IBM invented the floppy disk just to have a convenient media to ship patches to microcode.) Upon his return to Berkeley, he submitted a paper to IEEE Computer saying that the only way to build a VAX-class machine on a chip, with all its complexity in microcode, was to provide mechanisms that would allow patches to occur. The paper was rejected, as reviewers said such a technique was wasteful of silicon resources, which was certainly true, but it was also in Patterson's opinion not possible at the time to get the VAX microcode perfect in order to mass produce chips. If both positions were valid, then perhaps the solution was to re-examine the value of the instruction set complexity in the VLSI age?

The Berkeley work was done as part of a series of graduate classes, and the early statistics in the paper were generated by the first class investigating the RISC ideas, starting in January 1980. This series of courses included learning Mead-Conway design, investigating the RISC architecture ideas, and then implementing RISC-I. Many Berkeley students took some of the courses, but students who stayed all the way to the end included Dan Fitzpatrick, John Foderaro, Manolis Katevenis, Jim Peek, Bob Sherburne, and Korbin Van Dyke.

John Cocke of IBM visited for a day during those courses and gave us a very inspiring endorsement of our plan to keep the instruction set as simple as possible. Sometime that winter we also heard a talk by Forest Baskett in which he expressed his desire to have a really large number — e.g., "one thousand" — registers. Some time later, after we had completed the course series, Peter Denning came to give a "qualitative" architecture talk in Spring 1980, and he found an audience loaded with statistics as well as opinions about the value of complex architectures. The following year or two Nick Tredennick presented us with a seminar with the provocative title "Why RISC is a pile of junk." So these were rather exciting times!

One story of interest is where the name RISC came from. In the winter of 1980 Dave Patterson and Carlo Séquin were driving to Silicon Valley to go to a program committee meeting, and they were talking about what to name the project and where to get research funding for the project. Our hope was to get funding from the Department of Defense Advanced Research Project Agency, as DARPA was the prime agency for providing sufficient budgets to actually build chips and systems. The agency's motto was to fund "high risk, high payoff" proposals, as DARPA's responsibility was to push the state of the art to make sure the United States was not surprised technologically as it was by the Sputnik satellite in 1957. Hence we decided to name the project RISC since by definition it was "high risk", hoping that DARPA would see the high payoff and support our work!

One thing about the RISC vs. CISC debate is the revisionist history on what was CISC. The paper says the trend towards complexity was given by comparing VAX vs. PDP-11, iAPX-432 vs. 8086, System 38 vs. System 3. The VAX and the 432 were the ones that we questioned, and we think those concerns hold up pretty well today. The 8086 may have been inelegant, but it was not particularly complex, and this paper used it as the example of a simple machine. I think the trade press concluded that any commercial computer that wasn't a RISC must be a CISC, and hence the confusion.

Something to keep in mind while reading the paper was how lousy compilers were of that generation. C programmers had to write the word "register" next to variables to try to get compilers to use registers. As a former Berkeley Ph.D. who started a small compiler company said later, "people would accept any piece of junk you gave them, as long as the code worked." Part of the reason was simply the speed of processors and the size of memory, as programmers had limited patience on how long they were willing to wait for compilers.

This brings us to one final story about RISC a few years later. The register windows of RISC-I were there to make sure the operands stayed in registers versus in memory, as a RISC machine that keeps scalar variables in memory is a slow computer. When Patterson was consulting for Sun on SPARC, the Sun compiler expert was Steve Muchnick. Muchnick asked Patterson about register windows for Sunrise, the code name for SPARC. Patterson said variables need to be in registers, so the question was whether the Sun compilers were going to implement the graph coloring algorithm from IBM that did a very good job of register allocation. Muchnick got a ghastly look on his face, and gave an emphatic NO. Patterson said then sunrise better use register windows. Two years later when the machine shipped Sun's compilers had improved such that graph coloring was not such a ghastly prospect, but by then the die had been cast.

Looking at the technical content of the paper, the definition of RISC-I stands as a pretty good definition of RISC machines and the RISC-I instruction set is still a very reasonable 32-bit integer instruction set — given that you want to build the whole processor in less than 50,000 transistors. Now that we know about the 801, you can see the differences in the instruction format (16-bit immediate field vs. 13-bit for RISC-I) and terminology ("execute and branch" vs. "delayed branch".) The paper claimed the RISC-I code size was about 1.5 times the VAX, and that is probably still a reasonable estimate. Claiming that a single chip computer was faster than the best selling minicomputer was a bold claim, even if the claim was qualified by saying it was made on only using two small programs. Bhandarkar and Clark [1] did a careful study 10 years later when there were real systems to compare, and reached the same conclusion: "RISC as exemplified by MIPS offers a significant processor performance advantage over a VAX of a comparable hardware organization."

What have the authors and key students been doing since this 1981 paper?

David Patterson got his tenure shortly after this paper appeared and moved on to become department chair, and later SIGARCH chair and Computing Research Association chair. This paper described the first of a series of RISC-related research projects which produced several RISC chips: RISC-I, RISC-II, SOAR (Smalltalk On A RISC), and SPUR (Symbolic Processing Using RISCs), with RISC-I likely being the first VLSI RISC chip. In 1984 he started consulting with Sun Microsystems, which lead to SPARC being designed based on the various Berkeley RISC designs. In 1987 Randy Katz and Patterson developed Redundant Arrays of Inexpensive Disks (RAID), which showed how to get more performance and higher reliability from secondary storage. To help make the more quantitative approach to computer design more popular, John Hennessy and Patterson co-authored two architecture textbooks in the early 1990s. Since that time he has worked on Networks of Workstations (NOW) with Tom Anderson and David Culler, which built large-scale systems from smaller systems using off-the-shelf switched networks, and most recently on Intelligent RAM (IRAM) with Tom Anderson and Kathy Yelick, which integrates a processor into a DRAM chip to provide a small, fast, energy-efficient computer for mobile, multimedia applications.

Carlo Séquin was made Computer Science Division chair in the fall of 1980, and was thereby yanked out of the mainstream of the development of the successor chips to RISC I and II. When he returned to research full-time, after an influential sabbatical at Siemens Corp. in Munich, Germany, he decided to abandon chip building in favor of building tools that would ease the tedium of designing and debugging VLSI chips. Together with Richard Newton and Alberto Sangiovanni-Vincentelli they launched the Berkeley Synthesis Project which focussed a large number of faculty and graduate students on an effort to build CAD tools and integrated circuits in a symbiotic manner. Towards the end of the 1980's Séquin wanted to build CAD tools that would involve more than just the two dimensions used in chip layout. He had had a long-standing interest in Computer Graphics. The design and construction of Soda Hall, the new home of the CS Division since 1994, provided him with the opportunity to create a complete computer model of Soda Hall and to develop tools for the interactive exploration of virtual buildings. Since then he has worked on the development of CAD tools for architects and for mechanical engineers, and most recently even started to collaborate with sculptors who would implement his computer-generated artistic designs in wood or in bronze.

Dan Fitzpatrick got his Ph.D. in 1983, working on Computer Aided Design software for VLSI. He initially joined VLSI Research.

John Foderaro also got his Ph.D. in 1983, but his interest was in symbolic manipulation systems. He helped found a Berkeley software startup called Franz, Inc, which initially created a LISP programming environment. He still works for that company.

Manolis Katevenis was the lead graduate student on the project, and his 1983 dissertation won the ACM Distinguished Dissertation Award. He went on to work as a faculty member at Stanford university for a year before returning home to Greece. He is now with the University of Crete, Dept. of Computer Science, where he is currently Professor and Associate Chairman. Since 1985, he is also with the Institute of Computer Science, FORTH, Heraklion, Crete, where he is currently the Head of the Computer Architecture and VLSI Systems Division. His recent work has been on very fast switches (see http://www.ics.forth.gr/proj/avg.)

Jim Peek came to Berkeley for an MS, and has since worked for a variety of computer companies in Silicon Valley, and most recently was working at Sun Microsystems.

Robert Sherburne got his Ph.D. in 1984 and went to work originally at what became Pixar to build graphics hardware. He has since joined Silicon Graphics. Katevenis and Sherburne built RISC-II, which was probably the first microprocessor from a university accepted for publication at the prestigious International Solid State Circuits Conference.

Korbin Van Dyke also came for an MS, and joined an early 8086 clone company that was eventually purchased by AMD. He now works for Chromatics Research.

References

[1] D. Bhandarkar and D. W. Clark, "Performance from architecture: comparing a RISC and a CISC with similar hardware organization." *Fourth International Conference on Architectural Support for Programming Languages and Operating Systems,* pp.310-19, April 1991.