Architects Debate VLIW, Single-Chip MP

Or Something Completely Different—Reprogrammable Accelerators

by Linley Gwennap

Beyond the next generation of superscalar processors lies an undiscovered country, a place where today's rules of processor design may not hold true. At this year's Microprocessor Forum, three explorers gave us their visions of what this country will be like when, later this decade, we finally arrive. These visionaries told of titanic changes but, as is often the case, did not agree.

VLIW Promises Higher Performance

Wen-Mei Hwu, a professor at the University of Illinois, sees a bright future for very long instruction word (VLIW) processors *(see **080205.PDF**)*. This technology is reportedly being considered by Intel and HP as they construct their future instruction-set architecture *(see **080801.PDF**)*, but Hwu stressed that he does not speak for either of these two companies.

Hwu claims that future processors will be limited by current instruction sets. "If you want to construct a high-issue-rate machine without changing the instruction set, you will probably see something like a 10–20% return when you double the issue rate for the next generation. It's a sad truth, but that's the situation."

He explained that disruptions in the instruction stream defeat attempts to execute many instructions in parallel. "The real reason that you want VLIW is to deal with conditional branches and memory latency; you want the compiler to help.… If you are fetching eight instructions per cycle, you are going to run into branches left and right unless you have some kind of major change to your architecture. That's what's going to limit all these high-issue-rate machines in the near future. Even the four-issue machines are going to see this effect.

"But a VLIW compiler has the potential to do some good in this area," Hwu declared. He said that predicated execution—in which the execution of one instruction depends on the result of a previous instruction—can eliminate many branches, simplifying the instruction stream. He admitted that current architectures like PA-RISC, SPARC V9, and Alpha already have this feature (called "conditional move" in SPARC and Alpha), making it less clear what the benefit of VLIW would be.

To bolster his point, Hwu showed some performance simulations based on compiler technology that he claimed will be used commercially within the next year. Comparing a four-issue VLIW processor with a scalar processor on SPECint92 benchmarks, the results showed gains between 1.8× and 2.8× for the VLIW design; four-way superscalar processors typically gain less than 2× over scalar designs. For an eight-issue VLIW processor, Hwu expects gains of 4.0× to 5.6×, but these numbers assume "projected 1997 compiler technology."

Hwu also addressed some of the criticisms of VLIW. Many have noted that, in the 1980s, VLIW machines from Multiflow and Cydrome were commercial failures. Hwu made an analogy to the development of RISC: "By the time the industry people figured out how to do RISC right, the PA-RISC, PowerPC, and SPARC V9 instruction sets ended up to be very different from the RISC that the Berkeley people talked about a decade ago. So I wouldn't be surprised if, by the time VLIW processors come into the mainstream, they look completely different from the Multiflow and Cydrome machines."

For example, these early VLIW processors had no hardware interlocks at all, relying completely on the compiler for all instruction scheduling. Hwu believes that future VLIW devices will incorporate some interlocks, preventing the compiler from having to consider absolute worst-case timing.

These VLIW pioneers also made no attempt to be compatible with existing software. Hwu stated that "many future VLIW processors will, in some way, be upward-compatible with major current architectures. It would be insane for the x86 people to give up their compatibility advantage.… There will be some kind of interesting binary emulation technology to protect the existing software investment. In particular, you are going to see some extensive architectural support for binary emulation in most of these architectures."

Multiprocessing Is Easier to Build

Jeff Deutsch, of Deutsch Research, agreed that superscalar designs are reaching their limit. "I think the current superscalar processors are truly impressive, but getting much more complex is going to make the Apollo moon shot look like a weekend romp." He noted that run-time dependency checking of several instructions per cycle is very complex, as is managing superscalar and out-of-order execution. Finally, he concurs with Hwu that branches and dependencies will limit the return from increased issue rates.

But Deutsch also expressed serious concerns about VLIW, which he said requires a "magic compiler, but somehow the rabbit doesn't come of the hat." Predetermined instruction scheduling makes VLIW, in his eyes,

CLARENCE TOWERS

Figure 1. (left to right) Wen-Mei Hwu, Jeff Deutsch, and Nick Tredennick discuss future architecture techniques with moderator Michael Slater.

the "Moscow central-planning approach to instruction issue. In general, having more diversity and freedom to make decisions at run time tends to be more successful."

VLIW's static scheduling is inherently inefficient, he asserted. "Dynamic scheduling is good. Fast algorithms are fast because they take advantage of special cases. These special cases often involve run-time dependencies or run-time decisions."

So how should we solve the superscalar bottleneck? Deutsch noted that "there's been a solution available now for quite a few years that, unlike VLIW, has been commercially successful: symmetric multiprocessing (SMP). Sequent introduced a 32-processor server in 1984, and most major server vendors now offer SMP systems. Sun's SparcStation 10 is one of the most successful workstations around, and it goes up to four processors."

He pointed out that Windows NT, Novell NetWare, and most versions of UNIX support SMP systems or will soon. The challenge lies in creating applications that can take advantage of multiple processors, a process called multithreading. Some vendors are already shipping multithreaded software, mainly for CAD and transaction-processing applications. These vendors must rewrite their software by hand to take advantage of data parallelism, that is, the ability to perform the same process on multiple data points at the same time.

Hwu argued that this rewriting process is laborious and slow, and that it will be many years before a critical mass of multithreaded applications is available. Deutsch pointed out that all the next-generation RISC processors due next year support glueless multiprocessing, implying a boom in SMP system sales ahead. This boom should convince more software vendors to make the effort of multithreading their applications.

The ultimate expression of SMP is placing multiple processors on a single chip *(see **080605.PDF**)*. Deutsch said that this is a much simpler technique than highly superscalar designs and can scale to chips with hundreds of millions of transistors simply by adding more CPUs. In addition to integration, such a chip offers improved interprocessor communication compared with a discrete SMP solution: the interprocessor bus could be faster and wider, reducing the penalty when one processor accesses another's cache.

Reprogrammable Logic: A Way-Out Solution

Nick Tredennick, Altera's chief scientist, proposed a completely different approach: "If lack of parallelism in the instruction stream seems to be the problem, let's quit executing instructions." As a proof of concept, he pointed to the graphics accelerators and video accelerators that are widely used in PCs today; instead of executing instructions on the CPU, these chips execute entire functions such as bitBLT or YUV conversion, freeing the CPU to execute other tasks.

As special-purpose accelerators proliferate, however, it may make sense to replace them with a single block of reprogrammable logic. This block could be configured to perform a given task, then reconfigured for a different task after the first one completes. Ideally, this would save the cost of having multiple accelerators while increasing the flexibility of the system and avoiding the need to rewrite applications.

Having made this modest proposal, Tredennick proceeded to shoot it down. The biggest problem with reprogrammable logic today, he quipped, is "it's got more overhead than the federal government: approximately 125 transistors per usable gate." The reprogrammable accelerator would also require an immense amount of new software to make it work. Thus, Tredennick admitted that this idea is "even further out than VLIW."

So far, it appears that the two contenders to succeed superscalar designs will be VLIW and single-chip MP. The key issue is software support. VLIW advocates must demonstrate a static compiler that outperforms the dynamic scheduling done in hardware. For SMP to become pervasive on the desktop, software vendors must multithread their applications by hand until a parallelizing compiler appears. If neither side is able to deliver on its software promises, the onus will be on processor designers to squeeze more and more performance out of existing instruction sets and programming paradigms. ♦

