Bringing Parallelism Out of the Closet

Architects Debate Whether New Instruction Sets Are Needed

by Linley Gwennap

There is no doubt that new instruction sets are coming: the Intel/HP alliance has made this quite clear. At this year's Microprocessor Forum, a panel of leading processor architects discussed whether this change is inevitable and whether current instruction sets can survive the Intel/HP onslaught.

Moderator Michael Slater set the tone. "I think there's general agreement that if we take today's binaries, the path we're on is going to run into fundamental limits," he said. "That is, if we simply build processors with eight-instruction dispatch instead of four-instruction dispatch, and with four integer units instead of two integer units, and so on, we may get modest gains. But suppose you double everything again and again and again? No matter what you do with the hardware, if you're living with today's binaries, microarchitecture improvements alone won't provide another factor of 5 or 10.

"Today, we have instruction sets that have no way to convey parallelism information to the hardware, and we're trying to build microprocessors that dynamically, clock cycle by clock cycle, try to find and extract all the parallelism from the instruction stream in order to drive parallel execution units. Is that really the right way to do things? It would seem there is a benefit to having some way to convey to the hardware the parallelism information known by the compiler."

Advantages of a New Instruction Set

Bill Dally, a professor at MIT, amplified this point. "If you look at Pentium Pro, there is a philosophy…that parallelism is something you do in the privacy of your own execution unit, but you don't talk about it to the programmer. And that philosophy leaves a lot to be desired as we go to very large numbers of execution units," he said.

"We need to get the parallelism out of the closet and expose it to the programmer and make it visible, not just at the level of the instruction set but at the level of the execution engine. Instead of the reorder buffer deciding on the fly what can be issued in parallel over and over again, every time around the loop, the decisions should be made in software once, by the compiler."

Michael Mahon, an architect working on IA-64 at HP, agreed. "Ever-more-parallel superscalar machines are devoting more and more of the die to what I would characterize as administrative functions and, in particular, functions that need not be done repeatedly during execution but can instead be done by a compiler. We see an opportunity here, very much in the spirit of the original move to RISC, to use more of the die for producing results…

"A simple case is numerical computations, where loop unrolling is an important branch-elimination operation. If you attempt to unroll several times, to begin to cover things like L2 cache latencies or even—heaven forbid!—a memory latency, you have a demand for a lot more registers than are available in current instruction sets.

"Although renaming can be applied in heroic ways to overcome some of those problems," Mahon continued, "the compiler has a great deal more freedom to make the right things happen if it can actually allocate and name real registers that exist. This change also removes some administrative hardware [for register renaming] from the chip."

Taking the baton, Dally explained why the register renaming used in current processors is inferior to a large addressable register set. "If you try to do register renaming, there is still no covering…spills. If the compiler thinks it's run out of registers, it's going to spill them to and from memory and generate egregious amounts of memory traffic."

Can Current Instruction Sets Survive?

Given this stirring argument for new instruction sets, Slater asked the obvious question. "If a new instruction set offers considerable advantages, will that enable Intel to leapfrog all

Figure 1. At the Microprocessor Forum, a group of processor experts discussed prospects for achieving much higher performance in future CPUs. They are (l to r) consultants Brian Case and Andy Heller, HP's Michael Mahon, MIT's Bill Dally, Sun's Gary Lauterbach, Transmeta's Dave Ditzel, and moderator Michael Slater of MicroDesign Resources.

of today's architectures in performance? This could be a rather dire situation, since Intel is already thriving with what we could politely call a 'handicapped' architecture."

Sun's Gary Lauterbach, the lead architect for UltraSparc-3, gave a rebuttal, pointing out that other factors are limiting the performance of future microprocessors. "I don't believe we need to migrate to a new ISA [instruction-set architecture]; extensions to the current ISA will be sufficient. The ISA itself isn't particularly an inhibitor or an enabler of more parallelism…Really, the issues we must address in the next couple of generations are the wire delays on chip, that is, the communication between the units. These delays are a major impediment to building things like 12-wide pipelines.

"A second problem is in the memory hierarchy," he added. "Memory speed is not scaling with processor performance. We can add ISA extensions that should be sufficient to extract the parallelism necessary to address those issues." For example, the SPARC v9 architecture (see **070201.PDF**) implemented in today's UltraSparc includes prefetch instructions to help cover memory latencies.

Dave Ditzel is quite familiar with the strengths and weaknesses of SPARC, but he recently left Sun to found a new company, Transmeta. Perhaps foreshadowing the design of a future product, Ditzel opined, "I think new instruction sets are definitely necessary to significantly enhance performance, to expose more parallelism. But that doesn't mean you have to give up on the old instruction set as well….We're going to have to think about how old instruction sets can coexist with new hardware and new instruction sets."

Distribution Issues Could Be a Challenge

If several new instruction sets emerge, one challenge will be generating and distributing software that will function on a plethora of old and new processors. Standards groups have bandied about the idea of an architecturally neutral distribution format (ANDF) for years, but the concept never gained enough support to be adopted.

Brian Case, a long-time contributor to *Microprocessor Report*, thinks x86 may become the de facto ANDF. "I cringe when people say, 'Let's throw some x86 hardware in the corner here to execute the old instruction set.' Because a much cleaner way is to simply say, 'OK, we're going to move to a new instruction set, but our translation technology is good enough today that, although x86 is a very poor choice for an ANDF, it can be a source for our compiler.' "

Case explained, "The compiler can read the x86 code and translate it [to a new ISA], perhaps with a little bit of run-time emulation for those who insist on self-modifying code and things like that. This method will probably run x86 binaries faster than the best available x86 hardware." Indeed, Digital is using this technique today in its FX!32 translator (see **100302.PDF**).

Dally prefers a more elegant ANDF. "If you had a clean sheet of paper, which is never the case, what you would want is a distribution format that is the program-dependency graph out of the middle of the compiler."

But Case points out the real-world limitations of this strategy. "Why didn't ANDF catch on? One of the reasons is that application developers can't control the platform on which their applications will be run. This is a huge problem. You're talking about ballooning their support costs [to deal with a larger number of target platforms]."

Another Approach Is Multiprocessing

Other architects believe the way to address parallelism is not by changing instruction sets but by putting multiple processors on a single chip. As Andy Heller, former RS/6000 and Hal processor architect, pointed out: "You've got a thousand dimensions to get parallelism, but you guys keep focusing on one or two….[Dally] talked about adding more registers. But put five special-purpose processors to do all your I/O and all your DSP functions on the same chip, and you've now increased the number of registers fivefold without adding one new instruction. All you've done is put a multiprocessor on the thing. That's another dimension."

Heller noted this isn't exactly a radical idea in system design. "You've never had a PC that wasn't a multiprocessor. Tell me how you think you ran your display? Tell me how you think you ran your modem? You always had a multiprocessor. You just never liked to talk about it like that."

One problem with these legacy systems is that all the processors have different instruction sets. But perhaps this is a feature, not a problem. Ditzel suggested, "I don't think we should be constrained to having just one instruction set on a chip. For example, Intel added MMX. If you want to have a media processor, it's not clear to me that is the best way to do it. We're going to have enough transistors in the future to have an entire media processor on the CPU chip."

HP's Mahon argued that multiprocessing is simply another tool in the architect's box, not a replacement for highly parallel uniprocessors. "You obviously try to provide a fast interconnect [for] multiprocessors, but you also try to design the uniprocessor to be as fast as it can be, to handle the sequential parts of the computation as quickly as possible. There's not a straight tradeoff here.

"Some applications parallelize well, and some don't," he asserted. "You want to do both well. So I think it's incorrect to talk about multiprocessor parallelism, or multithreaded parallelism, as a *replacement* for instruction-level parallelism, since both are essential."

Over the next couple of years, the debate over new instruction sets will move from the theoretical to the real, as the first details of the Intel/HP collaboration emerge. Sun's Lauterbach, along with designers of PowerPC, MIPS, Alpha, and x86 processors, must either coax more speed from their aging instruction sets or trade in for a brand new one. Many factors will influence the decision, but as Heller pointed out, testosterone may tip the scales. The architect's credo: "If you want to be macho, create a new instruction set." M