

Why Change Instruction Sets?

New ISAs Promise Performance, But Others See Hope in Threads



The recent Microprocessor Forum provoked more debate than usual about the direction of high-end processors. Products rolling out in 2001 will deliver performance from varying combinations of instruction-level parallelism (ILP) and thread-level parallelism (TLP). These alternatives give new ammunition to the debate over whether new instruction sets are necessary to achieve high performance in the future.

Over the past several years, vendors have added new features to their high-end microprocessors, virtually in lock-step, embracing first superscalar, then instruction reordering as the key techniques for improving performance. The x86 vendors even adopted internal RISC engines, minimizing microarchitectural differences among processors.

Intel and its IA-64 partner HP were the first to break free from the throng, proclaiming that next-generation instruction sets will be needed to fuel continued performance increases. Even Intel admits that changing instruction sets will cause some disruptions for end users, but it insists that the pain is necessary for those users to gain the maximum processor performance that IA-64 offers.

The focus of IA-64 is on increasing ILP by improving the interface between the compiler and the hardware, letting each work more effectively in scheduling instructions. With this approach, the compiler can directly control hardware resources such as large register files, branch predictors, the memory hierarchy, and a plethora of function units.

For business reasons as well as technical reasons, other vendors want to stay with their current instruction sets. They simply do not have the resources to gain software support for their own next-generation instruction sets, and Intel isn't licensing IA-64. Even IBM, which has a full Intel patent license, doesn't have access to key IA-64 patents that have been assigned to a holding company, IDEA, that is jointly owned by Intel and HP.

For these vendors, sticking with their current instruction set allows them to seamlessly serve their existing installed bases, a lucrative business. In the case of IBM and Compaq, they are playing both sides of the fence by selling IA-64 systems as well as their own RISC systems, so they can keep their customers happy either way.

But IBM and Compaq have a problem: if they can't keep their in-house RISC processors competitive with IA-64 in performance, their customer bases will gradually migrate to IA-64. This migration will reduce RISC system revenue

that can be invested to develop new RISC processors. If that happens, their RISC lines will eventually fall behind.

At the Forum, IBM and Compaq unveiled their plans to keep pace with IA-64's performance. Both are aggressively pushing ILP, but they have added a new weapon: thread-level parallelism. IBM's Power4 will exploit TLP using two physical CPUs per chip, while the Alpha EV8 will, through the wonders of simultaneous multithreading, have four virtual processors per chip.

This new weapon should be an effective response to IA-64's server performance. Most server applications today have a number of software processes, or threads, that can be assigned to individual processors, and these applications run effectively on systems with 4, 8, or even 64 processors. Thus, TLP exploits a proven method of increasing server performance.

TLP is less effective in smaller systems, where only one or two threads do most of the work. Unix systems typically have many processes running at any given time, but frequently only one is stressing the CPU, while the others handle simple background tasks. Windows 98 doesn't even support multiprocessing, making TLP moot in today's PCs.

These situations are changing. More modern workstation applications are designed to run effectively on two or more processors by breaking themselves into parallel threads. As a result, most workstations today support two or more processors. Even some PC applications are now multi-threaded, and Windows 2000 will support multiple processors in PCs.

Future workloads will accelerate these trends. Multimedia looks to be the biggest performance driver in future PCs, and these applications are inherently full of thread-level parallelism. In addition, the Java programming language makes it easy to develop multithreaded applications. As multimedia and Java become more popular, the amount of TLP in workstations and PCs will rise.

These trends bode well, allowing RISC architectures to keep pace with IA-64 by exploiting thread-level parallelism. Even x86 could reach IA-64 performance levels in a future multiprocessor chip from AMD (see MPR 10/25/99, p. 24). The Intel/HP approach should still hold an advantage on single-thread benchmarks, but as TLP becomes more prevalent on the desktop, the value of this advantage will diminish. □