Architects Look to Processors of Future

Applications, Instruction Sets, Memory Bandwidth Are Key Issues

For this special issue, we asked several processor architects how, based on 25 years of history, they see the microprocessor continuing to evolve in the future. Their responses discuss several technical barriers to success and how they might be overcome. Equally important is an often overlooked issue: what will people do with all this performance?

GORDON BELL

Many New Applications Will Emerge

In 1947, the big idea (perhaps of all time) was the stored program computer that was soon to operate. In the same year, the transistor, a second and equally big idea, was invented. By the mid 1960s, a way of fabricating and interconnecting transistors on silicon substrates was invented and in use.

The development of the microprocessor in 1971 ensured the evolution of computing would continue in a very focused fashion. The next 15–25 years look equally bright. The only form of intelligence more easily, cheaply, and rapidly fabricated is the human brain, estimated to have a processing power of around 1,000 million million ops/s (one petaops), with a memory of 10 terabytes [Cochrane, 1996].

For five decades, hardware has stimulated the evolution of computer platforms of various performance, size, cost, form, and applications, from watches and pacemakers to mainframes. It is safe to predict computers in 2047 will be at least 100,000 times more powerful. If hardware continues to evolve at the annual factor of 1.6 we know as Moore's Law [Moore, 1996], computers that are 10 billion times more powerful will exist! Magnetic-storage density and fiber-optic data transmission rates have evolved at the 60% rate (a doubling every 18 months, or 100 times per decade), too.

It is also likely that, since improvements in algorithms and methods often occur at the same rate as in hardware, any future goal is likely to be reached in half the time one would predict based on hardware alone. I don't believe the homely computer, built as a simple processor/memory structure, will take on a very different look, but rather will continue on an evolutionary path of only slightly more parallelism of instruction execution. For the past decade, real application performance (RAP) of microprocessors has diverged from the peak announced performance (PAP) that follows Moore's Law. This trend will continue!

Figure 1 shows past hardware evolution and a 50-year forecast of the future. The next 15 years, based on semiconductor progress, are most likely to follow this trend. After that time, the figure shows a diverging range of possibilities.

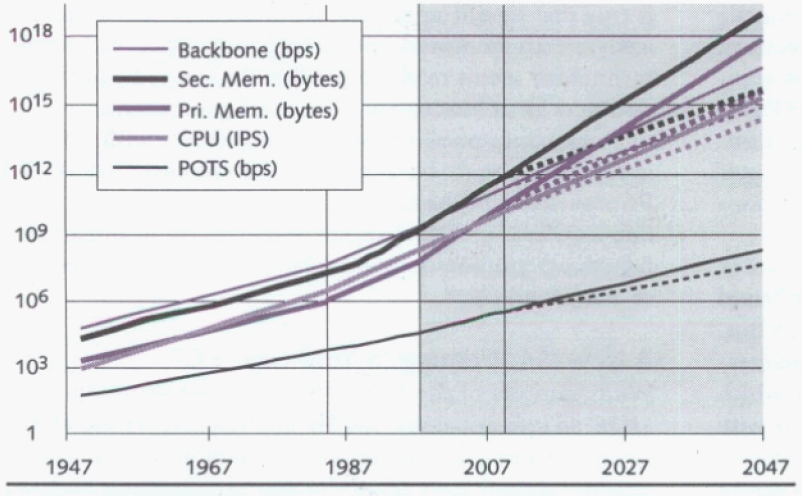

Figure 1. Aggressive projections (solid lines) show processing power, memory, and bandwidth increasing rapidly throughout the next 50 years. Even in the conservative case (dashed lines), there will be enormous improvements in the future. (Source: Gordon Bell, 1996)

What Forms Will the Future Computer Take?

All intellectual property and everything bitable will be in cyberspace. With cyberspace, the speed limit is our ability to find new places. Bitability comes from the hardware and software interfaces (I/O) that the computer has acquired, created, or evolved to allow it to communicate with people and the physical world. We eventually expect speech, video, and gesture interfaces, followed by having computers that anticipate. Surely, we can expect a "do what I say" metaphor within a decade, since it has been a dream for so long.

Direct body interfaces are increasingly important, including touch, direct nerve stimulus, and artificial organs, eyes, ears, and limbs. I don't expect computers will interface by taste and smell. For achieving the mobility and navigation in the physical world that would enable useful robots, the big inventions already exist today as demonstrations, with video recognition, global-positioning systems (GPS), laser sensing, single-chip phased-array radar, and sonar. They have to evolve to low-cost components and become fast enough. By 2047, I expect homes, commercial areas, and factories will have useful robots that do not require extensive training.

New computer classes based on price will

Dependency hints. Part of the hardware that supports speculative out-of-order execution is logic that checks for dependencies between pending operations in the reservation station(s). Dependency information in the instruction stream can be provided by an optimizing compiler and could simplify implementations by eliminating the need for hardware to recompute interoperation dependencies.

Effect on Instruction Sets

An instruction set combining most of the traditional RISC tenets with the above ideas would be the basis for a powerful architecture that would increase the efficiency of a processor implementation, but some sacrifices would be required.

One sacrifice would be increased dependence on compiler technology. To even reasonably exploit a machine incorporating all the ideas listed above would require a relatively sophisticated compiler. This is not a severe deficiency, however, because a sophisticated compiler is already required to even approach the performance potential of existing high-end microprocessors.

Perhaps the biggest sacrifice would be compromises in the instruction format. To combine three-address, register-to-register operations (a basic RISC tenet) with a large number of registers and guarded execution would require four bit-hungry register specifiers. With 128 registers, the four register specifiers alone consume 28 bits. To encode a reasonable number of operations would require more than 32 bits for basic arithmetic instructions.

One solution is to step backward to two-address, destructive operations. With one less register specifier, a 128-register machine would still have 11 bits in a 32-bit format for encoding basic arithmetic operations.

If instructions longer than 32 bits are acceptable, it might make sense to step backward in another way and create complex instructions that—don't faint—combine arithmetic operations and memory references. It is also possible to encode two arbitrary and possibly dependent arithmetic operations in a single long 64-bit instruction. The precedent for this can be found in an architecture proposed by compiler-researcher Bill Wulf several years ago. (SuperSPARC could internally cascade two dependent operations in a single cycle because its ALUs were disproportionately faster than its caches.) Another instruction that can benefit from a long format is the multiway branch.

In a machine with some instructions longer than 32 bits, it probably makes sense to allow two or three different instruction formats, perhaps 64 bits, 32 bits, and 16 bits. This concept is a logical extension from RISC-like architectures such as NEC's V800 (see MPR 10/25/93, p. 25), which intermixes 32- and 16-bit instructions. An architecture with multiple instruction sizes complicates the implementation of instruction fetching and decoding, but given current technology—and the Pentium Pro and AMD K5 as proof of concept—the complexity is not overwhelming. With just two or three power-of-two instruction sizes, the design challenges should be only a fraction of those encountered in an x86 implementation.

Multithreading Support Can Be a Big Win

Potentially one of the biggest parallelism-discovering wins is support for multiple threads of execution within a single program. By definition, separate threads are independent and therefore inherently parallel. By fetching instructions from separate threads, a processor can easily find parallelism and keep its execution resources busy. While one thread is stalled waiting for a load instruction to return a value from memory, the independent instructions from another thread can be executing.

Multithreading support requires, at least conceptually, multiple program counters, one for each thread. Thus, architectural support for multiple threads might benefit from some changes to the instruction set, but the changes are in the form of additional instructions, not modifications to the fundamental instruction semantics or formats. Thus, it should be possible to add multithreading support to existing architectures, but it may make more sense to add it to a new architecture where the integration into the instruction set and implementation can be as clean as possible.

The first implementation of the MicroUnity architecture (see MPR 10/23/95, p. 11) has built-in multithreading to accommodate its extreme pipelining.

Existing Architectures Will Still Thrive

While the new features listed above are many and impressive, the current state of the art in superscalar design defines a clear path to steadily improving the performance of existing architectures over the next few years. Increasing the width of the fetcher and decoder, the size of the window (reservation station) of pending instructions, and the number of execution units should allow the processor to find more independent instructions and execute more instructions per cycle. It is true that significantly more hardware will be required to achieve each modest increment in performance, but circuit technology seems ready to provide the needed additional resources for at least two or three more generations.

Also, the appeal of binary compatibility with existing applications cannot be overstated. The entrenchment of the PC standard and the fact that customers continued to buy Sun machines over the past few years despite an embarrassing price/performance ratio support the costly development of compatible chips.

A New ISA: Coming in Your Next PC?

While samples of the Merced chip are not expected until 1H98, all indications are that HP and Intel are serious about releasing a new architecture. Since most of the non-x86 CPU vendors already have relatively new architectures and are fighting for market share, it seems unlikely that a new architecture will emerge from any of the RISC vendors. Thus,

Continued on page 27

continue to be determined by applications and their resulting markets together with three factors: hardware platform technology (e.g., semiconductors, magnetics, and displays); hardware/software interfaces to connect with the physical world, including people; and network infrastructures (e.g., the Internet and eventually home and body area networks).

My theory of computer class formation, based solely on using lower-cost components and different forms of use to stimulate new structures, accounted for the emerging of minicomputers (1970s), workstations and personal computers (1980s), and personal organizers. The World Wide Web has stimulated other computer classes to emerge, including network computers, telecomputers, and television computers that are combined with phones and television sets, respectively. As Table 1 shows, this basic theory also accounts for the emergence of embedded and low-cost game computers using worldwide consumer distribution networks. Mobility via a radio network opens up more future possibilities that are not just adaptations of cellular phones.

Generation	Platform (logic, memories, O/S)	User Interface	Network Infrastructure
The beginning (direct and batch use)	Vacuum tube, transistor, core, drum and mag tape	Card, paper tape	None originally; computer was self-contained
Interactive timesharing via commands	Integrated circuit, disk, multiprogramming	Glass, teletype and keypunch, command language	POTS using modem; proprietary nets using WAN
Distributed PCs and workstations	The microprocessor. PCs and workstations, floppy, disk, distributed O/S	WIMP (windows, icons, mouse, pull-down menus)	WAN, LAN
World Wide Web	Evolutionary PCs and workstations, servers everywhere, Web O/S	Browser	Fiber optics backbone, WWW, HTTP
SNAP (Scalable Network and Platforms)	PC uni- or multiprocessor commodity platform	Server provisioning	SAN (System Area Network) for clusters
One dial tone: phone, videophone, TV and data	Network computer, telecomputer, TV computer	Telephone, videophone, television	xDSL for POTS, cable, fiber (longer term); home area nets
Do what I say	Embedded in PCs, hand-held devices, phone, PDA	Speech, common sense	Body area nets. IR and radio LANs for network access
Anticipatory by observing user needs	Room monitoring, gesture	Vision, gesture control, common sense	Home area nets
Robots	No special	Radar, sonar, vision, mobility, arms, hands	IR and radio LAN
Ubiquity embedded	$1–$100 devices that interoperate	Computer-to-computer control	Home area and body area networks

Table 1. New computer classes and their enabling components. (Source: Gordon Bell)

Within a few years, scalable computing, using an arbitrary number of commodity-priced computers and commodity high-speed networks to operate as one, is likely to replace traditional computers, i.e., servers of all types! We call this approach to computing SNAP, for scalable network and platforms [Gray, 1996]. The underlying parallelism is a challenge that has escaped computer science for decades.

As communication instruments, computers enable the substitution of time and place of work, creating a flat, equal-access world [CNRI, 1996]. After nearly 30 years of the Internet, people-to people communication via e-mail and chat remains the top application. Is telepresence for work, learning, and entertainment the long-term "killer app"?

Can these systems be built in this short time? Will computers interface with humans biologically, rather than in the superficial, mechanical way they do now? More likely, nearly-zero-cost communicating computers will be everywhere, embedded in everything from phones and light switches to all-seeing, all-changing pictures. They'll be the eyes and ears for the blind and deaf, and they will eventually drive vehicles.

We will need to be fully connected anywhere at all times. The big idea is fiber-optic cable that evolves to carry more bits per second each year at 1.6× per year. Perhaps an equally big idea is in the making: the high-speed digital subscriber link, a.k.a. "the last mile," that permits high-speed data to go to the home via the world's trillion dollars of installed copper connections. In parallel, radio links will enable "anywhere computing." Body and home area networks are parts of the network story that need to be invented.

As VP of R&D at Digital for 23 years, Gordon Bell led the development of the PDP and VAX minicomputers and other products. He is now a senior researcher at Microsoft and can be reached at gbell@microsoft.com.

References

CNRI. "Vision of the NII: Ten Scenarios," Reston, Virginia, 1996. See also *www.cnri.reston.va.us.*

Cochrane, Peter. Many papers on the future of computers; *www.labs.bt.com/people/cochrap.*

Gray, J. "Scalable Servers"; *www.research.com/research/barc.*

Moore, Gordon. "Gigabits and Gigabucks," University Video Corp. Distinguished Lecture, 1996; *www.uvc.com.*