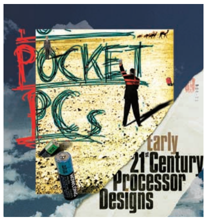
Early 21st Century Processors



The computer architecture arena faces exciting challenges as it attempts to meet the design goals and constraints that new markets, changing applications, and fast-moving semiconductor technology impose.

Sriram Vajapeyam
Indian Institute
of Science

Mateo Valero
Technical University
of Catalonia

The excitement and challenges of computer design lie in meeting multiple design goals and constraints while riding the dual horses of fast-improving technology and changing application domains. The target market, available technology, and target applications impose the design goals and constraints.

The architect attempts to build a machine that leverages particular technology and performs well on popular and important applications within market constraints—and this results in the dual horses setting some of the race's constraints. Since semiconductor technology improves rapidly, and the popularity and importance of applications can change, the design constraints take on the form of moving targets when computer design time—or time to market—becomes significant with respect to technology and application changes. Often, these factors force a computer design project to change its target market, target application, underlying semiconductor technology—or all three—in midstream.

Because computer architecture research is very closely coupled to these real-life issues, technology breakthroughs and the inherent difficulties of predicting market factors and novel applications can take researchers by surprise.

COMPUTER ARCHITECTURE ARENA

We can appreciate the difficulty of predicting the future when we consider that all four major players in the computer architecture arena can affect each other, to a greater or lesser degree, as Figure 1 shows.

Semiconductor technology creates markets, enables and constrains architectures, and makes new applications possible. Markets can, in turn, affect technology. For example, a high-revenue market such as for high-end servers will funnel more effort into technology development suitable for that market. The importance or popularity of specific applications and application domains can impact technology development—as frequently happens with military applications, for example. Popular *architectures* can, to a lesser extent, affect development of suitable technology. An example is the impact of the von Neumann architecture on polarizing memory (DRAM) and processor technologies, resulting in the slow development of novel integrated logic-DRAM technology processes.[1]

Application characteristics directly impact architecture—no architecture runs all applications optimally, so developers tune architecture design to target applications. An application's popularity can create new markets as well—consider the impact of Web browsers or online transaction processing.

Markets impact architecture in terms of cost, performance, size, power, and time-to-market goals. Markets can also trigger new applications—for example, the embedded market, enabled by cheap technology, can trigger novel applications. Thus, with every player affecting every other player, the computer design arena is indeed complex and its trends hard to predict.

FUTURE PROCESSORS

Processor architecture and microarchitecture have now entered an era of multidimensional changes in their operational environment. The underlying semiconductor technology continues to improve significantly, although some questions have arisen about the continued validity of Moore's law,[2,3] which states that single-chip transistor counts double roughly every 18 months. Technology improvements have led to newer

0018-9162/01/$10.00 © 2001 IEEE

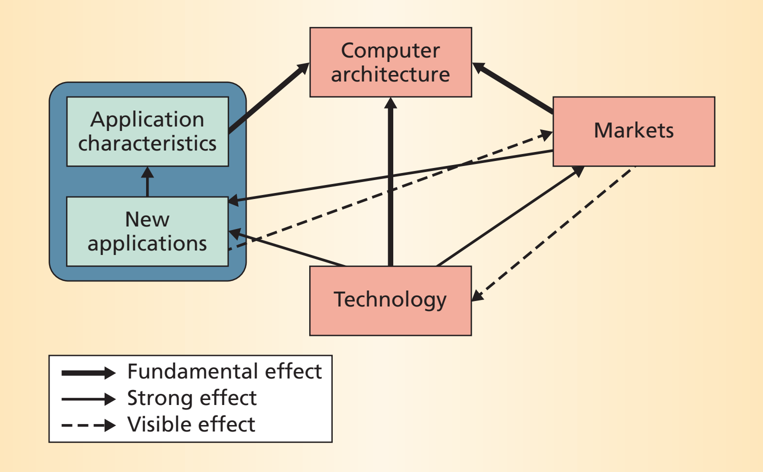


Figure 1. The computer architecture arena. The four major players—applications, architectures, technology, and markets—all affect each other to a greater or lesser extent.

microarchitectural design constraints of power dissipation, localized communication (due to relatively slower wires compared to faster logic), and design and verification complexities. These constraints are a direct result of the fundamental trends of increasing transistor density (leading to more transistors on a single chip) and increasing switching speeds (enabling faster processor clock rates). Continuous reduction in transistor sizes brought about by improved silicon processes fuels these trends.

The improvement in semiconductor technology, along with concomitant reductions in cost per transistor, has triggered new markets and application domains, especially embedded and mobile computing. The use of chips in embedded and mobile applications has resulted in further market constraints on power dissipation, space (size), and cost-performance. In parallel, traditional computer markets have also undergone significant changes. While commercial transaction processing and Web servers dominate the high-end computing market, multimedia and digital signal processing applications are beginning to dominate the desktop and mobile-computing markets. These applications exhibit behavioral characteristics that are quite different from the traditional general-purpose applications as represented, for example, by the SPEC CPU2000 benchmark.

This special issue attempts to capture some of this excitement in early 21st century processor architecture through articles that discuss design directions for future processors. We also feature several sidebars that capture snapshots of interesting current commercial processors and prototypes. Early versions of the main articles of this special issue were presented at a session organized by the guest editors at the 7th International Conference on High-Performance Computing (HiPC 2000, http://www.hipc.org) held in Bangalore, India, in December 2000.[4]

POWER: A NEW TECHNOLOGICAL CONSTRAINT

A significant impact of improving semiconductor technology has been the power dissipation by a chip, which has increased rapidly because both the number of transistors on a chip and the switching frequency (processor clock rate) have grown quickly. A single high-speed processor can consume 100 watts of power while being significantly smaller than a corresponding light bulb.

Given this rapid increase in power, in "Power: A First-Class Architectural Design Constraint" on pp. 52-58, Trevor Mudge argues that we should consider power dissipation as a first-class design constraint for general-purpose processors. Power has been accepted as a design constraint much earlier in the embedded processor arena because of the product constraints under which developers deploy them. High-end supercomputers such as the Cray-1 had to grapple with heat dissipation problems more than 25 years ago, and designers resorted mainly to cutting-edge and costly cooling techniques because performance was a paramount consideration.

Recently, processors have begun to trade off performance (or the processor clock rate) for power, while using operating system, microarchitecture, and circuit techniques to minimize power consumption. In the "Microarchitecture Uses a Low-Power Core" sidebar on page 55, Mike Morrow describes how Intel's XScale processor provides knobs to the system designer to help control power. Interestingly, the power factor could well contain the commercial marketing frenzy for higher processor clock rates.

NOVEL ARCHITECTURES FOR TRADITIONAL APPLICATIONS

The burgeoning transistors on a chip have had a direct impact on processor architecture.[5] Harnessing these many transistors requires keeping design and verification efforts in check. Interestingly, the newer multimedia applications that contain enormous parallelism can often consume vast numbers of transistors architected together in relatively simple ways.

For traditional applications, however, researchers have taken two different approaches. The straightforward approach simply increases the sizes or number of on-chip caches and other processor resources, and even places multiple processors on a single chip or allows various forms of multithreading. A more aggressive form attempts to place memory (DRAM) on the same chip as the processor.[1] The straightforward approach is effective for a significant fraction of potential applications. At the same time, it neither requires nor generates novel architectures, and it is not effective for all applications. Simple enlargement of traditional processors runs into the previously mentioned wire-length barrier as well.

A second approach has been to use novel modular and hierarchical architectures for harnessing the transistors to improve the performance of traditional single-thread applications. In addition to addressing the complexity issue, this hierarchical approach also addresses the relatively slower communication delays in denser chips. This approach has resulted in interesting and novel architectures.

A major microarchitectural trend has been toward hardware (and runtime software) that is devoted not to direct execution of the program, but to monitoring and learning the program's execution characteristics and subsequently recasting the program for faster execution. Starting points for such metahardware included the trace cache proposal,[6,7] which captures instruction sequences in dynamic execution order, and the trace processor proposal,[8] which recasts traces at runtime as well as organizes the processor into a hierarchy of modules partitioned at traces.

The ILDP paradigm that James E. Smith discusses in "Instruction-Level Distributed Processing" on pp. 59-65 assumes similar modular, distributed processors based on simple, fast processing elements and includes an explicit accounting of communication costs in the microarchitecture and possibly even in the instruction set. The ILDP paradigm includes metahardware—helper engines—that can monitor, recast, or optimistically execute program segments. Variations of trace-based and early ILDP techniques have reached the commercial marketplace, for example, the HAL UltraSparc V processor[9] and the Intel Willamette/Pentium 4.[10]

In parallel with the metahardware approach, researchers have also explored approaches that devote the extra transistors simply to running system software that performs the monitoring and recasting functions.[11] The virtual-machine technology discussed in the ILDP article offers an intermediate approach in which a layer of specialized hardware-software sits between the application and the underlying ILDP hardware. This virtualization layer both manages the underlying distributed hardware and efficiently monitors and recasts the executing software.

The combination of increasing hardware resources, memory latency, and throughput-oriented workloads has resulted in a significant investment in multithreaded and chip-multiprocessor (CMP) architectures. An interesting unified architecture would behave as either a multithreaded processor or a single-thread processor on a per-application basis. It appears that researchers can build such a processor in fairly straightforward ways, except for the question of how a single thread would use all the multithread resources.

Several researchers are exploring the creation of multiple slave threads that aggressively use the additional resources to prefetch data, precompute some results, or preidentify control-flow paths for the slower main thread—perhaps in a speculative manner. In "Speculative Multithreaded Processors" on pp. 66-73, Guri Sohi and Amir Roth outline a framework for categorizing such threads. The sidebars in this article—"Piranha: Exploiting Single-Chip Multiprocessing" by Luiz André Barroso and colleagues and "Cray MTA: Multithreading for Latency Tolerance" by Burton Smith—illustrate two current multiprocessing approaches: a CMP targeted at transaction processing workloads and a multithreaded, multiprocessor architecture targeted at high-ILP and highly parallel applications.

Embedded computing's market and product constraints have resulted in new combinations of design constraints, such as very good specialized performance at very low power and very low cost.

EMBEDDED COMPUTING: THE NEW FRONTIER

The exponential reduction in cost per transistor (for volume production) that has occurred with increasing chip densities has resulted in a novel market and application space: embedded and mobile computers. The embedded market is already bigger and growing faster than the PC market, and the mobile market is expected to become bigger than the PC market soon. Embedded computing's market and product constraints have resulted in new combinations of design constraints, such as very good specialized performance at very low power and very low cost.[12] Technically, this often translates to fast, specialized computing using very small memories that have no backup secondary storage.

In "Embedded Computer Architecture and Automation" (pp. 75-83), Ramakrishna Rau and Mike Schlansker describe the design constraints for embedded processors and make a strong case for the automation of computer architecture in this domain. They argue that the exploding embedded market will require many different specialized designs, well beyond the industry's architecture and VLSI design capacities. This, they claim, will lead to a novel dimension of computer architecture—automation rather than manual design.

While automation is not a panacea, it should work well within specific domains that have well-cataloged architectural choices. In the "PICO Architecture Synthesis System" sidebar on page 80, Ramakrishna Rau outlines a prototype software system that can pick optimal architectures from a choice library for specific target applications.

BACK TO THE FUTURE

Clearly, computer architecture faces a new era. On some fronts, it appears as if we might be heading back to the future. For example, multimedia workloads share many characteristics with vector applications that date back three decades. Embedded computing's resource constraints resemble those of bigger computers from many years ago. However, we will see these constraints often in conjunction with a whole new set of other constraints. So we need to use the lessons of the past together with new discoveries.

As with several previous efforts,[5,12,13] we dare not make strong predictions. Given the arena's complexity, we wonder whether architecture research should be more decoupled from the marketplace so that research results become available well before we encounter specific real scenarios, despite the difficulties of predicting such scenarios. Decoupling would also guard against research becoming bogged down by current, short-term constraints.

We can easily recall a time when it would have been difficult to carry out or publish research on memory-efficient programs—transistor budgets were increasing anyway, and few foresaw the advent of embedded computing, which makes memory size a critical factor. Similarly, the commercial constraint of binary compatibility perhaps killed a significant amount of architectural research—again, few foresaw the current binary translation and virtual machine technologies.

Perhaps it makes sense for computer architecture research to follow the path of more established and mature sciences. To wit, we could make interesting assumptions about different technology and application characteristics—independent of their immediate validity or possibility—and then pursue interesting and novel solutions to the problems that such scenarios pose. When not taken to its own extreme, this approach could result in less incremental research, more emphasis on new and significant ideas—research leading the industry rather than competing with or, worse, repeating it—and faster progress overall. Certainly the time seems ripe in 2001 to foster a new odyssey of imagination, excitement, and riches in computer architecture. ✷

References

1. C.E. Kozyrakis and D.A. Patterson, "A New Direction for Computer Architecture Research," *Computer*, Nov. 1998, pp. 24-32.
2. *International Technology Roadmap for Semiconductors,* Semiconductor Industry Assoc., San Jose, Calif., 2000; http://www.semichips.org.
3. S. Hamilton, "Taking Moore's Law into the Next Century," *Computer*, Jan. 1999, pp. 43-48.
4. M. Valero, V.K. Prasanna, and S. Vajapeyam, eds., *Proc. 7th Int'l Conf. High-Performance Computing* (HiPC 2000), Lecture Notes in Computer Science, Springer-Verlag, Heidelberg, 2000; http://www.springer.de.
5. D. Burger and J.R. Goodman, "Billion-Transistor Architectures," *Computer*, Sept. 1997, pp. 46-49.
6. E. Rotenberg, S. Bennett, and J.E. Smith, "Trace Cache: A Low-Latency Approach to High-Bandwidth Instruction Fetch," *Proc. Int'l Symp. Microarchitecture*, IEEE CS Press, Los Alamitos, Calif., 1996, pp. 24-34.
7. A. Peleg and U. Weiser, "Dynamic Flow Instruction Cache Memory Organized around Trace Segments Independent of Virtual Address Line," US patent 5,381,533, Patent and Trademark Office, Washington, D.C., 1994.
8. S. Vajapeyam and T. Mitra, "Improving Superscalar Instruction Dispatch and Issue by Exploiting Dynamic Code Sequences," *Proc. Int'l Symp. Computer Architecture,* ACM Press, New York, 1997, p. 12.
9. M. Shebanow, "Sparc 64 V, A High-Performance System Processor," presentation, Microprocessor Forum, Nov. 1999; http://www.hal.com/products/.
10. G. Hinton, "Next-Generation IA-32 Processor Architecture (Willamette)," presentation, Intel Developer Forum, Feb. 2000; http://developer.intel.com.
11. E.R. Altman, D. Kaeli, and Y. Sheffer, "Welcome to the Opportunities of Binary Translation," *Computer*, Mar. 2000, pp. 40-45.
12. M. Schlett, "Trends in Embedded-Microprocessor Design," *Computer*, Aug. 1998, pp. 44-49.
13. Y. Patt, "Identifying Obstacles in the Path to *More*," *Computer*, Dec. 1997, p. 32.

***Sriram Vajapeyam** is a faculty member of computer science at the Indian Institute of Science. His research focuses on processor architectures and memory systems, with a significant recent contribution being trace processors. Vajapeyam received a PhD in computer science from the University of Wisconsin-Madison. Contact him at sriram@csa.iisc.ernet.in.*

***Mateo Valero** is a professor of computer architecture at the Technical University of Catalonia. His research interests are in computer architecture, with special interest in processor organization, memory hierarchy, and compilation techniques. He is a Fellow of the IEEE and a member of the ACM and the Spanish Academy of Engineering. Contact him at mateo@ac.upc.es.*