Computer Design Starts Over

Erik P. DeBenedictis, Sandia National Laboratories

To maintain Moore's law, the semiconductor industry decided a decade ago that a new transistor was imperative. That silver bullet has yet to materialize, but computer design innovations are now maintaining or even exceeding expected scaling progress. This theme issue gives a cross-sectional view of these new scaling drivers.

0018-9162/17/$33.00 © 2017 IEEE

Five architecture-based articles in this theme issue provide what I see as the most up-to-date view on the future of computing. We know that semiconductors were phenomenally successful in driving the IT sector to large profits, so when the improvement rate for transistors slowed, it was only natural to try to replicate previous success through transistor-like alternatives. Instead, entirely new avenues of computing research have emerged that seem able to keep Moore's law going using the same transistors.

THE MOVE BEYOND CMOS THAT DID NOT HAPPEN

In April 2008, the International Technology Roadmap for Semiconductors' (ITRS's) panel of experts assembled in Koenigswinter, Germany. This meeting was to be the commitment point for "Beyond CMOS," the name set aside for the successor to current transistorized logic that was expected to support many generations of additional scaling. ITRS was an organization sponsored by semiconductor industry associations worldwide to plot the course of Moore's law, which at that time meant larger memories and faster microprocessors. At this meeting, the leadership group, the International Roadmap Committee (IRC), challenged my fellow participants and me to identify one to two devices or logic families and report on them at the next ITRS meeting that December in San Francisco. This was the touchpoint for roadmapping industry's transition to the new technology.

The sought-after device was not available then, nor is it today, but what is really striking is that the computer industry was betting its future on the hope of a discovery. A report was dutifully delivered at the December 2008 meeting, but it only recommended further research on carbon-based electronics, such as carbon nanotubes and graphene. What followed was a decade of chaos from which a new industrial structure is just now emerging.

As we know, a suitable drop-in replacement for the transistor has yet to be found for computer logic—although, as discussed in this month's Rebooting Computing column, "Sustaining Moore's Law with 3D Chips" (vol. 50, no. 8, pp. 69–73), 3D memories have been developed and are in use. Microprocessor companies continue to release new processors, though they do not work much better than their predecessors. Despite the fact that there has been a lot of trauma for semiconductor companies, we really do not have a lot to complain about from the standpoint of the economy. At this writing, the top five largest companies in the world by market cap are in the IT sector, up from none in 2008. However, these companies all started out in the Internet business, not semiconductors, and they are headed toward new business in AI. We need a way to reconcile the failure to develop the new transistor with the fact that the IT industry was successful in the interval.

KILLER APPS BEAT FASTER MICROPROCESSORS

"Killer apps" have always driven the big computing expansions, but the semiconductor industry's flagship product, the microprocessor, is just an enabler for applications. Killer apps bring automation to whole categories of work once done by humans, such as the word processor in the 1980s and the web browser in the 1990s. Although experience might lead us to believe that semiconductor innovations expand the economy, this is not the whole story; improved semiconductors enable new applications, but it is the applications that provide value to the end users.

Predicting the future is notoriously risky, but if we accept mainstream news coverage as true, then neural networks, self-driving cars, machine learning, and other applications under the AI umbrella are candidates for the next killer apps. These new applications learn their behavior rather than having it programmed into them in the traditional sense, so future killer apps might not need a microprocessor, dethroning the flagship product of the past but allowing the industry to grow with a different product.

HOW TO GET PERFORMANCE?

The April 2008 meeting revealed industry's focused plan for improving computer performance: find a replacement for the transistor. There was no corresponding strategy for changing the architecture of the microprocessor or higher levels of the computer technology stack. Since the replacement for the transistor has yet to materialize, we now must consider other paths.

When viewed as isolated devices, today's transistors are within an order of magnitude of the physical limits of energy efficiency at typical computer speeds. This could be why no one has found a dramatically better replacement. However, the laws of physics that govern transistor performance allow energy efficiency to be traded off for speed. The articles in this issue explore this topic.

IN THIS ISSUE

The five articles in this issue collectively define a research program that could expand the IT sector over

time. If microprocessors had continued their exponential performance-improvement path, the innovations described in these articles would not have been necessary. So these research results represent true alternatives to the Beyond-CMOS devices rather than developments that would stand in parallel with them.

Self-driving cars

In "Computer Architectures for Autonomous Driving," Shaoshan Liu, Jie Tang, Zhe Zhang, and Jean-Luc Gaudiot describe an architecture clearly distinct from the microprocessor along with a rationale for its necessity in empowering one of the leading killer app candidates, self-driving cars. Alan Turing's Turing machine could compute any computable function based on a precise and broad mathematical definition of computability, and John von Neumann's ubiquitous architecture can compute any computable function because it is "Turing complete." However, these theories only cover the computer's ability to do many different things, not their ability to do them quickly or efficiently. A self-driving car must identify hazards in time to avoid hitting them, and the control system must be energy efficient enough to fit within the power budget of the automobile. If microprocessors were to continue improving exponentially, all we would have to do is wait, but as this is no longer happening, we need to invent a new architecture.

Liu and his colleagues propose a computational architecture for autonomous driving comprising a microprocessor or a von Neumann processor, a digital signal processor, a GPU, and a field-programmable gate array. Each component would be dedicated to a specific subtask of autonomous driving for which it excels, and the article accompanies each pairing with an analysis showing how the combined architecture meets the speed and power limits necessary for an autonomous vehicle.

Exploiting tradeoffs in speed and energy efficiency

In "Energy-Proportional Computing: A New Definition," Rathijit Sen and David A. Wood explore ways to make existing microprocessors more effective by exploiting physics-based speed–power tradeoffs in the underlying transistors. The laws of physics—and the principles of overclocking known to hackers—show how to get more performance from a microprocessor by increasing the clock rate above the manufacturer's limits. The supply voltage must be increased as well, causing a sharp decrease in energy efficiency. The reverse is called undervolting, and it raises energy efficiency. Many computers experience a natural change in load across intervals, such as day versus night. By adjusting systems software, existing microprocessors could be overclocked during periods of high demand and undervolted during periods of low demand, thereby optimizing the tradeoff between system capacity and energy costs. Until a few years ago, energy efficiency was ably handled though Moore's law, thus the involvement of architects is only now important.

Additional tradeoffs including parallelism

In "Voltage, Throughput, Power, Reliability, and Multicore Scaling," Fei Xia, Ashur Rafiev, Ali Aalsaud, Mohammed Al-Hayanni, James Davis, Joshua Levine, Andrey Mokhov, Alexander Romanovsky, Rishad Shafik, Alex Yakovlev, and Sheng Yang use software to control speed–power tradeoffs in more ways. If a computer's load comprises many small tasks, the method described in the previous article by Sen and Wood can optimize the speed of each processor to match its load at a given instant.

However, important problems arise where a computer system has just one or a few tasks. The tasks can be split up via parallelization, but the effectiveness of parallelization varies by the nature of the underlying algorithm and the amount of programmer time and effort that can be devoted to recoding. This leads Xia and his colleagues to a more complex analysis, where the voltage and clock rate of a microprocessor are varied along with switching the algorithms to use more or fewer processors. The additional degree of freedom over the method in Sen and Wood's paper allows the optimum point to be better.

Larger systems rather than faster ones

In "Scaling the Computer to the Problem: Application Programming with Unlimited Memory," Ike Nassi describes an approach that is somewhat the reverse of those discussed by Sen and Wood and Xia and his colleagues. Namely, Nassi explores a way to make computers larger, rather than faster and more energy efficient. Computer applications are defined by a program, which is software. The program puts different demands on the computer hardware when solving larger problems. In some cases, programs will get bogged down because the problem needs more processor cycles than the processor was designed for, even though the processor is running efficiently. In other cases, the computer

ABOUT THE AUTHOR

ERIK P. DEBENEDICTIS is a technical staff member at Sandia National Laboratories' Center for Computing Research. His research interests include the future of computing, computing for spacecraft, cybersecurity, and superconducting electronics. DeBenedictis received a PhD in computer science from Caltech. He is a member of IEEE. Contact him at epdeben@sandia.gov.

system does not have enough memory. Most of today's computers avoid crashing in this case by swapping data between memory and disk on the fly, but this method comes with a large performance penalty. A computer actually stops productive work and waits while data is swapped between disk and memory, and the computer runs so slowly you wish it had crashed.

Manufacturing a computer that can support a lot of memory requires wider address busses and more pins in the chip, imposing a cost on the majority of users, who mostly run word processors or other applications that do not need much memory. As a result, computers are designed to handle "average" applications with margin to support applications needing up to, say, 10 times the average amount of memory. These limits impede research, particularly given the increased attention to data-intensive computation and related fields like AI; that is because these applications could need 100 times the average amount of memory. Nassi's article describes how to support these large applications using software that fuses regular processors.

The visionary final article

At the beginning of this introduction, I described industry's 2008 mandate to put the transistor's successor on the path toward production. Prior to introducing the last paper, it could be useful to compare the beliefs in 2008 with the collective impact of the four articles described above.

In 2008, computer companies produced microprocessors and memories with designs that were popular at the time, such as Pentium, IBM Power, DDR (for memory), and so on. These designs abstracted the properties of transistors into simple parameters like the speed and energy of logic gates. Architectures were abstracted to instruction sets or bus protocols. The advantage of a drop-in replacement for the transistor is that nobody would have to "mess with" these abstractions, thereby simplifying management of the industry.

The previous four articles each violate the underlying abstractions in some way. However, these four articles are incremental, meaning they accept most conventions and change just one or two in order to create a plotline for their articles.

In the final article, "The Weird, the Small, and the Uncontrollable: Redefining the Frontiers of Computing," Christof Teuscher presents a long-term vision: computing is not only an engineering discipline with a product history but also the science of what can be computed, how quickly, and with how much energy. Teuscher considers how you might build a computer from scratch, setting aside most or all conventions. The process in this article opens up more degrees of freedom and, presumably, would yield a more effective result.

As evidenced by the articles in this issue, I realize we did not correctly answer the question in December 2008 about which one or two Beyond-CMOS devices should go into production and appear in products, well, about now. We should have said, "No such device can exist due to constraints placed on it by the abstractions of computer logic and the microprocessor architecture in which the devices are expected to operate. Instead, the right course is to have a diverse set of researchers chip away at the conventions of computing, changing the environment in which transistors are applied so they can be more effective."

Four of the articles in this issue show the result of this work, which should be sufficient for a number of generations if applied properly.

Although the transistor would seem to have a lot of life left in it, it is unlikely to be the ultimate answer. As pointed out in Teuscher's article, there are many other ways to compute aside from transistors. Some of these are likely to be more effective, although more time might be required to be refine these ideas and move them through the manufacturing process.

ACKNOWLEDGMENTS

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the US Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.