
Computing has always been quick to evolve because of changes in semiconductor devices and technologies—no more so than at the peak of the high-performance computing (HPC) pyramid. But the current general consensus is that HPC is in its last generation, and a crisis is coming. True, computing has weathered crises before: submicron’s lithography challenges and the end of Dennard scaling were addressed with device-physics and computer architecture changes that enabled computing performance to continue to increase exponentially.

But now, just when those tricks are all but tapped out and Moore’s law as the industry knows it is ending, demands for performance in excess of 1 million trillion floating-point operations per second (1 exaflops) are arising from novel software paradigms to address problems in big data, machine learning, and analytics that tax existing computational capabilities. The resulting pressure on computing creates a perfect storm that constitutes crisis with a capital C. Indeed, many industry experts believe that, by 2020, computing will reach the long-predicted performance wall.

What new computing or semiconductor technologies are needed to restart computer performance progress? That very question inspired Tom Conte and Elie Track to propose the IEEE Rebooting Computing Initiative (RCI;
rebootingcomputing.ieee.org) in 2012. Concurrently, the International Technology Roadmap of Semiconductors (ITRS) led by Paolo Gargini had been researching and defining the elements of a new technology roadmap, ITRS 2.0 (www.itrs2.net), that reached beyond dimensional scaling. In 2014, the RCI decided to partner with ITRS 2.0 to build on the roadmap and give it the broader scope of defining future computing. The partnership culminated in the May 2016 launch of the IEEE International Roadmap for Devices and Systems (IRDS; standards.ieee.org/develop/indconn/irds/index.html), which RCI is continuing to develop under the aegis of the IEEE Standards Association in consultation with the IEEE Computer Society.

With the launch of the IRDS program, IEEE is taking the lead in formulating a comprehensive, end-to-end view of the computing ecosystem, including devices, components, systems, architecture, and software. As the “The Rebooting Computing Initiative’s Spreading Influence” sidebar describes, the RCI has been shaping future computing through summits that identify potential solutions and their challenges. Now the IRDS is set to deliver a 15-year vision that encompasses systems and devices, setting a new direction for the future of the semiconductor, communications, networking, and computer industries. Bringing the IRDS under the IEEE umbrella will define the next Moore’s law of computer performance and accelerate the delivery of novel computing technologies in the market.

MOORE’S LAW VERSUS COMPUTER PERFORMANCE

To define the way forward, the industry must deeply understand how they relied too heavily on Moore’s law to spur exponential increases in computer performance. From the mid-1960s, society has taken these increases for granted, and initially that faith seemed justifiable, as computers got faster, cheaper, and more efficient with each generation. But such accelerated progress cannot go on indefinitely, and the end is now in sight. The signs have been there some time: exponential performance scaling stalled a decade ago at the transistor level, although multicore architectures temporarily concealed the effects from end users.

So what happened? In retrospect, it is easy to see how events conspired to end exponential progress in computer performance even though Moore’s law has continued and will in some form persist. Trying to dissociate Moore’s law from exponential performance increase is uncomfortable for many computer designers because the two ideas have become inextricably linked in the industry. Quite simply, Moore’s law will continue because the semiconductor industry will continue to deliver more transistors per dollar, as Moore predicted. However, exponential performance increases will not
continue because the newly delivered transistors will be no more efficient than those from the prior generation.

**Organization to fit von Neumann’s vision**

In 1945, John von Neumann described his computer architecture as a processing unit containing an arithmetic logic unit and several registers, a control unit containing an instruction register and program counter, and memory units for data and instructions. Access to a large external memory storage unit was also part of the overall structure. The sequential ordering of instructions in his architecture enabled programmers to write and debug complex software in a relatively straightforward manner, but made the computers’ speed directly proportional to the speed of the underlying parts: first vacuum tubes and then transistors. Computers would get faster as device speed increased—but speed was never the central thesis of Moore’s law.

In 1965, Gordon Moore wrote a paper that predicted many things about semiconductor progress, but the most notable was that the number of usable transistors would double annually with design evolution and technology improvements and that, in 10 years, as many as 65,000 transistors would be available to design a product. Moore’s assertion made transistor abundance a resource for system designers: they could exploit growing transistor availability to create more tantalizing offerings. The industry rose to that challenge; by the time Moore made his second prediction in 1975, that the number of transistors would double every two years, Silicon Valley was home to more than 40 companies.

Moore’s observation about the number of transistors per chip was not the only important trend being enunciated during the 1970s. Between 1972 and 1974, Robert H. Dennard formalized geometric scaling—a set of equations that predicts electrical properties of transistors under scaling. The consequence of Dennard’s equations was that transistors would get a lot smaller, but just a little faster.

**Reorganization to gain efficiency**

Because transistors were getting only a little faster, their speed could be but part of the performance picture. Another part was to introduce parallelism, which required architectural innovation. IBM’s Project Stretch began in 1960 with the goal of studying ways to enhance computing performance by changing the computer’s organization, essentially establishing computer architecture as a discipline. The company continued to drive architectural novelty with Gene Amdahl’s idea of separating microarchitecture from the instruction-set architecture. IBM showed that multiple models of the IBM 360 could be made with different microarchitectures—all capable of running the same software without recompilation. What the industry knows as cache memory was introduced in the IBM 360 model 85.

While IBM explored pipelined execution through instruction-level parallelism and out-of-order execution in the IBM 360 model 91, Control Data was doing likewise in the CDC 6600. These techniques enabled computer performance to grow while maintaining the illusion of von Neumann’s one-instruction-at-a-time architecture.

**Reliance on pipelining**

Until the early 1980s, high-performance, mainframe, and miniature computers were built from discrete transistors or through small-scale integration. The PC revolution was the next major shift, combining computing with semiconductors and ushering in the idea of pipelined superscalar microarchitectures.

**From PC to superscalar architecture**

PC designers discarded discrete transistors in favor of an industry-standard architecture and OS, made possible in part by the availability of microprocessors—monolithic silicon CPU implementations. In the early to mid-1980s, the semiconductor and computer industries entered an unusual arrangement in which the vast majority of manufactured PCs used microprocessors from Intel and software from Microsoft. The “market” of two companies became one in which silicon technology, instruction-
set architecture, and software architecture were defined and locked into backward–forward compatibility. Indeed, only one viable avenue remained for the computing ecosystem to progress: take a page from earlier lessons and enable multiple architectures to run the same software. This strategy’s aim was to keep improving x86-based microprocessors at a biannual pace, which required complex microarchitectures similar to those in the IBM 360 model 91 and the CDC 6600. Major advances, such as UC Berkeley’s High-Performance Substrate (HPS) microarchitecture and the Intel P6 project, allowed microarchitectures to work behind the scenes to execute instructions in parallel while keeping the programmer’s view of an x86 architecture. These advances produced pipelined superscalar microarchitectures, which are still in use.

Origins of the power wall. With these new microarchitectures, designers could increase clock frequency, the $f$ parameter, with each generation—but at a price. A high-level relationship binds the key electrical parameters of any technology: dynamic power per transistor is proportional to frequency of operation times the square of the operating voltage ($P \sim V^2f$). Thus, the degree to which operating voltage could reduce power increase was limited by the transistors’ operating threshold voltages. The implication of this limitation is that the total dynamic power dissipated by an entire IC can be expressed as $P \sim Nf$, where $N$ represents the total number of transistors operating simultaneously. From this simplified expression, it is obvious that increasing the number of transistors at Moore’s law pace and increasing the operating frequency by 15 to 20 percent per generation was bound to reach a thermal dissipation limit—the power wall.

Moreover, the same superscalar microarchitectures that enable higher frequencies though deeper pipelines have a nasty habit of wasting work. The more pipeline stages per processor, the more instructions need to be worked on at once to keep those stages busy. However, the instruction supply is bounded by branch instructions, so the hardware must wait for branch instructions to finish before it knows what to execute next. To keep the pipeline full leads to speculative execution: hardware predictors using what they learn from past program behavior to predict what branches will do. Even though an individual prediction’s accuracy remained constant, microprocessors discarded a progressively higher fraction of their work as scaling deepened pipelines.

Pipelining, speculative execution, and operating frequency kept on increasing through the 1990s until, by 2003, processors exceeded 200 W per chip. This milestone marked the crossing of a power threshold that requires far more expensive cooling technologies, which were outside the system-cost envelope of PC hardware at that time. The industry had to choose which to slow down: the growth of the microprocessor’s transistor number from one generation to the next, or the operational frequency rate. They decided on the first option, which maintained Moore’s law but sacrificed frequency growth. As Figure 1 shows, increasing operating frequency as the main tool to increasing computing performance was no longer viable.

Power wall consequences
The number of transistors per IC is expected to grow at Moore’s law pace for the foreseeable future, but any strategy to scale down transistors in favor of higher operational frequency will have to deal with the power wall. Thus, transistors could easily operate beyond 100 GHz, but the number of transistors in a single IC would not increase at Moore’s law pace.

So far, the industry has not come up with a satisfactory response to either the speed or power problem. Multicore architectures, one attempted solution, require substantial software rewriting.
OUTLOOK

FIGURE 2. Four future computing approaches and the extent to which they disrupt the traditional computing stack (left). At the far right (level 4) are non–von Neumann architectures, which completely disrupt all stack levels, from device to algorithm. At the least disruptive end (level 1) are more “Moore” approaches, such as new transistor technology and 3D circuits, which affect only the device and logic levels. Hidden changes are those of which the programmer is unaware.

PROPOSED WAYS FORWARD

Attendees at the RCI’s four past summits reached consensus on the idea that any solutions to extending computing performance and efficiency would have to radically depart from the straightforward interpretation of Moore’s law. As a 2015 IEEE Spectrum article put it,\(^5\)

> Today’s technology makes a 1-exaflops supercomputer capable of performing 1 million trillion floating-point operations per second almost inevitable. But pushing supercomputing beyond that point to 10 exaflops or more will require major changes in both computing technologies and computer architectures.

To address that requirement, the RCI meetings covered a range of solutions to the impending end of current computing paradigms, which can be characterized in terms of disruption to the computing stack, as Figure 2 shows.

Non–von Neumann computing

The most radical approaches rethink computing from the ground up, and will require new algorithms, languages, and so on. Chief among these is quantum computing, which uses properties of quantum mechanics to solve problems in optimization, search, and whole number theory. Although a quantum computer can be used as a universal computing platform, it will be no better than a conventional computer outside a limited set of problems. However, the quantum computer’s advantage is so large for some of those problems that it has the potential to shake the foundation of conventional scientific, engineering, business, and security practices. For example, a working quantum computer could factor the product of two large primes in a nanosecond,\(^6\) which undermines asymmetric-key encryption. This encryption standard, which is central to every facet of e-commerce and national security, is based on the notion that such factoring is computationally intractable.

Another non–von Neumann approach is neuromorphic computing, which leverages what is known about the human brain’s operation to create new computing technologies. Neuromorphic computers do not attempt to replicate the brain, but rather draw from the neuroscientific aspects that enable humans to solve problems with great efficiency, such as recognizing and classifying patterns in text, audio, or images. Neuromorphic computers can be simulated on modern computers, but the true energy efficiencies come from specialized hardware built specifically for the task.

Neuromorphic algorithms differ greatly from traditional algorithms and overlap the important discipline of machine learning. The industry can now simulate neuromorphic
computers on top of traditional computers, but the devices and circuits in neuromorphic computers with energy efficiencies as high as the human brain are not applicable to traditional computer construction. On 20 October 2015, the RCI released a white paper in response to a White House Office of Science and Technology Policy (OSTP) request for information to establish a new grand challenge in computing. This in part led to the OSTP issuing a new grand challenge: create a new type of computer that can proactively interpret and learn from data, solve unfamiliar problems using what it has learned, and operate with the energy efficiency of the human brain. This challenge will require a fundamental rethinking of how computing is done because existing paradigms are far too power inefficient.

Architecture changes
Architecture changes will require new programming. Although old software can run on these architectures, the software cannot exploit the new, more energy-efficient features. One architecture change will be to deploy specialized accelerators that perform a given function such as fast Fourier transform in a very energy-efficient way. Part of the energy-efficiency savings comes from these specialized units being nonprogrammable, and thus foregoing the energy consumed by a general-purpose processor’s instruction fetch and decode hardware.

Another change will stem from the complementary approaches of approximate and stochastic computing, which aim to remove the waste that results when computers calculate results to higher-than-required accuracy and precision. Although this combination can save significant power and improve computing speed, practical application will require new algorithms, which in turn will demand modified programming languages and architectures.

Hidden changes
Hidden changes are architecture changes transparent to the programmer. Changing the microarchitecture allows existing programs to take advantage of technologies that will appear after conventional Moore’s law is redefined because the interface between the instruction-set architecture and microarchitecture guarantees software compatibility. Techniques that aim to build novel computer microarchitectures while maintaining software compatibility include adiabatic computing, reversible computing, and cryogenic superconducting computing. Adiabatic computing reduces power dissipated in resistors by temporarily storing power in inductors and capacitors instead. Although it is applicable to conventional CMOS technology, resulting power reductions might be somewhat limited. Reversible computing can in principle go further, to completely eliminate power dissipation. Cryogenic superconducting computing, which uses low-temperature superconducting materials, has received significant funding from the Intelligence Advanced Research Projects Agency (IARPA) because of its potential in creating a superconducting computer (www.iarpa.gov/index.php/research-programs/c3). A successful cryogenic quantum computer must overcome several significant obstacles, including the lack of design automation tools, insufficient flows to handle large designs, and the limited focus on manufacturing. Current superconducting devices are being produced at feature sizes many generations behind standard CMOS logic. An alternative to adiabatic and cryogenic computing is to use lower-quality transistors. Today’s transistors are designed to be as reliable as possible, which limits their operating voltage. But power is proportional to the square of voltage, so relaxing transistor reliability could allow for much lower-power designs. However, to realize this, the industry needs new microarchitectures that can detect and repair the occasional computational error.

Ultimately, any changes in microarchitecture, function unit design, logic, and devices will need to be large to reap any significant gain in power efficiency and speed.

More “Moore”: new transistor technology
The approach least disruptive to the computing stack is to extend the last few generations of computers that embody Moore’s law. Because current transistor technologies cannot both be power efficient and operate reliably at the extreme scales that future technology will demand, a new transistor technology is needed. This is not trivial, but two likely successors to the transistor are tunneling field-effect transistors (TFETs) and carbon-nanotube field-effect transistors (CNFETs).

Moore’s law originated in the era of 2D chips, but the extension to 3D is obvious. Fabricating 3D memory is particularly promising, because memory categorically generates little heat per device and so the shift to 3D is less likely to produce cooling problems. Some new memory devices can be constructed by evaporatively
depositing devices and wires on top of logic circuits—currently to 64 layers, but likely more layers will eventually be possible. This “monolithic 3D” approach should produce new 3D circuit architectures by 2020, which will have memory collated with logic deposited right on top of it. 3D circuit architectures can drastically reduce interconnect delays, reduce power, and speed up communication between logic and memory levels. In addition, most of these new memories are nonvolatile, offering an additional stand-by power reduction.

**RODMAPPING THE INDUSTRY**

All novel computing hierarchies require study and then formidable work to transition them from the lab to practice. The RCI reached consensus that a roadmap of the way forward would, among other goals, have to identify mountains the industry must surmount to get to computing’s new era. Building on ITRS 2.0 was the first move toward creating the IRDS, but adding new roadmapping focus teams was a necessary next step to an in-depth exploration of future systems and devices.

**Building on ITRS 2.0**

From the mid-1970s to the end of the 1990s, the semiconductor industry grew with geometrical scaling as the main method to decrease transistor cost while improving performance and increasing number. Gargini, then Intel’s director of technology strategy, realized that the problems of reengineering the MOS transistor (in terms of both manufacturing process and structure) and interconnect lines were impacting the global as well as the US semiconductor industry and, in 1998, he promoted and launched the ITRS in association with organizations from the EU, Japan, Korea, Taiwan, and the US.

The ITRS identified the imminent end of Dennard scaling and outlined a set of revolutionary innovations aimed at continuing the semiconductor industry’s historical trends. Called equivalent scaling, this paradigm supported the industry’s growth from 2000 to the present and became foundational to ITRS 2.0, which was established in 2012 to align IC development with the new electronic industry ecosystem. The next decade will see 3D power scaling become the dominant method to continue and exceed historical trends and provide new IC architectures.

**IRDS focus teams**

The shift from ITRS 2.0 to the IRDS in May 2016 involved adding two new, top-level focus teams of leading experts: the applications benchmarking (AB) team and the systems and architectures (SA) team. Figure 3 shows how these two teams interact to guide the rest of the roadmapping activity.

The AB focus team is comprised of volunteers from industry (currently IBM; Microsoft Research; Renesas, Inc.; EEMBC; AMD; Hewlett-Packard Enterprise; and ARM, Ltd.), government (Sandia National Laboratories), and academia (the University of Westminster, UK; the University of Illinois at Urbana–Champaign; and Georgia Institute of Technology). Membership is open to others. The AB focus team maintains a list of key computing applications that can be viewed as grand challenges. Table 1 shows the initial list. Each application area in Table 1 can be important in multiple computing markets. The SA team supplies this list to the AB team, which creates a cross matrix, such as that in Table 2, with market drivers along the top and application areas along the left side. The matrix shows which application areas are important for each market driver (X), which are critical “gating” applications (G), and which are power sensitive (P).

The SA team then takes both the cross matrix and the performance roadmap of application areas that the AB team produced and generates a list of system metrics understandable to the rest of the IRDS focus teams. For example, for the smartphones market driver, the SA team creates a roadmap of the required number of application
processor cores, graphics cores, total package count, display megapixels, sensors, and antennas. Some of these parameters depend on the architectures being used. For example, multiple architectures can perform feature recognition. For each architecture, SA creates a set of requirements in terms of device speed, bandwidth requirements, frequency requirements, and so on.

The ultimate contribution of this front-end roadmapping will be a deep understanding of what device types are required in future computing. The industry has emphasized only devices for the traditional logic in microprocessors and memory devices. This emphasis must switch to identifying and understanding the devices needed for analog computation, superconducting, quantum computing, and approximate and stochastic computing, among others. Thus, activity to support the IRDS is integral to driving...
OUTLOOK

FIGURE 4. Future computing poses challenges (in red) for system software. Problem definition, for example, will require problem-definition languages that are independent of the computation platform. Selecting a computing platform will require making an intelligent decision about which platform will most efficiently execute the problem and then compiling the problem definition into executable code for the chosen platform. Significant challenges remain in creating new OSs and virtualization for these very different computing approaches.

Problem definition
When the entire computing stack is no longer present, how does software define the problem to be solved? If even the algorithm could change according to the platform used to solve the problem, there must be a way to express that problem that is platform independent. Some algorithm selection can be hidden from the user through APIs, but not all of it. In general, the very nature of expressing a problem to a computer will need reexamination.

Platform selection
This area has a twofold challenge: how to select the architectural platform for finding a solution, and then how to compile the program for that platform. The selection problem is similar to traditional OS problems, but with much-higher-level information available to the selector. Instead of knowing that a certain process requires so much memory and CPU time, the selector will know the kind of computation needed and must then choose the most efficient architectural platform.

Similarly, compilation will be more than translating from a high-level language to machine language. The first problem is that no abstract low-level intermediate code will be able to span machine languages as diverse as those from neuromorphic to quantum computing. The second problem is that the computational platform might use entirely different algorithms for the same problem. Again, APIs will hide this complexity from the user—for example, by employing optimized solutions from a library of code for the platform—but code libraries will not include all possible routines, thus new code-optimization problems will emerge.

OSs and virtualization
OSs for architecture approaches at disruption levels 3 and 4 (see Figure 2) have yet to be designed. Resource management and multiprogramming for non–von Neumann architectures such as neuromorphic and quantum are not yet understood. Similarly, virtualization has proven to be a key technology for reliability and load balancing in today’s datacenters. But an open question is how to virtualize so that a neuromorphic or quantum program can move to another architecture if the chosen hardware has failed. Answering such questions will require significant research and development.

The RCI has been working since 2012 to find a path forward for computing. Its most recent activity—the first IEEE International Conference on Rebooting Computing (icrc.ieee.org) in October 2016—expanded the discussion of issues and brought the idea of rebooting computing to more academic, industry, and government research communities. In parallel with the RCI’s efforts, the IRDS is roadmapping a future for computing that the industry can follow to make the next computer revolution a reality.

REFERENCES

**ABOUT THE AUTHORS**

**THOMAS M. CONTE** is a professor of computer science and of electrical and computer engineering at the Georgia Institute of Technology. His research interests include novel computing architectures, parallel computing, and neuromorphic computing. Conte received a PhD in electrical engineering from the University of Illinois at Urbana–Champaign. He is an IEEE Fellow, cochair of the IEEE Rebooting Computing Initiative (RCI), and vice chair of the IEEE International Roadmap for Devices and Systems (IRDS). Contact him at tom@conte.us.

**ERIK P. DEBENEDICTIS** is a staff member at the Center for Computing Research of Sandia National Laboratories. His research interests include advanced computing and quantum computing. DeBenedictis received a PhD in computer science from Caltech. He is a member of the IEEE Computer Society, ACM, and the American Physical Society and cochair of the RCI. Contact him at epdeben@sandia.gov.

**PAOLO A. GARGINI** is chair of the IRDS and has been chair of the International Technology Roadmap for Semiconductors (ITRS) since its inception. He received PhDs in electrical engineering and in physics from Università di Bologna. Before retiring, he was Intel’s Director of Technology Strategy and Fellow of the Technology & Manufacturing Group. He is an IEEE Fellow and an International Engineering Consortium (IEC) Fellow. Contact him at paologargini1@gmail.com.

**ELIE TRACK** is CEO of nVizix. His research interests include superconducting electronics, devices, and computer architectures. Track received a PhD in physics from Yale University. He is an IEEE Fellow and cochair of the RCI. Contact him at elie.track@nvizix.com.

---

**myCS** Read your subscriptions through the myCS publications portal at [http://mycs.computer.org](http://mycs.computer.org)

---

**IEEE Computer Society magazines and Transactions are now available to subscribers in the portable ePub format.** Just download the articles from the IEEE Computer Society Digital Library and you can read them on any device that supports ePub. For more information, including a list of compatible devices, visit [www.computer.org/epub](http://www.computer.org/epub).