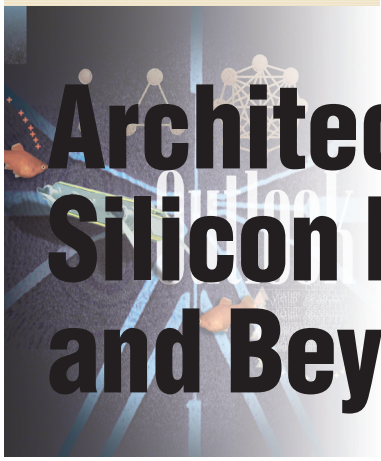


Architectures for Silicon Nanoelectronics and Beyond

R. Iris Bahar, Brown University
Dan Hammerstrom, Portland State University
Justin Harlow, University of South Florida
William H. Joyner Jr., Semiconductor Research Corp.
Clifford Lau, Institute for Defense Analyses
Diana Marculescu, Carnegie Mellon University
Alex Orailoglu, University of California, San Diego
Massoud Pedram, University of Southern California

Although nanoelectronics won't replace CMOS for some time, research is needed now to develop the architectures, methods, and tools to maximally leverage nanoscale devices and terascale capacity. Addressing the complementary architectural and system issues involved requires greater collaboration at all levels. The effective use of nanotechnology will call for total system solutions.

The semiconductor industry faces serious problems with power density, interconnect scaling, defects and variability, performance and density overkill, design complexity, and memory-bandwidth limitations. Instead of raw clock speed, parallelism must now fuel further performance improvements, while few persuasive parallel applications yet exist.

A candidate to replace complementary metal-oxide semiconductor (CMOS) technology, nanoelectronics could address some of these challenges, but it also introduces new problems. Molecular-scale computing will likely allow additional orders-of-magnitude improvements in device density and complexity, which raises three critical questions:

- How will we use these huge numbers of devices?
- How must we modify and improve design tools and methodologies to accommodate radical new ways of computing?
- Can we produce reliable, predictable systems from unreliable components with unpredictable behavior?

The effective use of nanotechnology will require not just solutions to increased density, but total system solutions. We can't develop an architecture without a sense of the applications it will execute. And any paradigm shift in applications and architecture will have a profound effect on the design process and tools required.

Researchers must emphasize the complementary architectural and system issues involved in deploying these new technologies and push for greater collaboration at all levels: devices, circuits, architecture, and systems.

WHAT IS NANOARCHITECTURE?

We define *nanoarchitecture* as the organization of basic computational structures composed of nanoscale devices assembled into a system that computes something useful. Nanoarchitecture will enable radically different computational models, and, due to its potential for large capacity, might also provide superior capabilities in some areas. Since architecture is rarely created in a vacuum, these issues will greatly affect nanoarchitecture development.

There are two paths to follow: evolutionary and revolutionary.

Evolutionary path

Silicon semiconductor technology will continue to shrink. But there's an increasing performance gap between device technology and its ability to deliver per-

0018-9162/07/$25.00 © 2007 IEEE

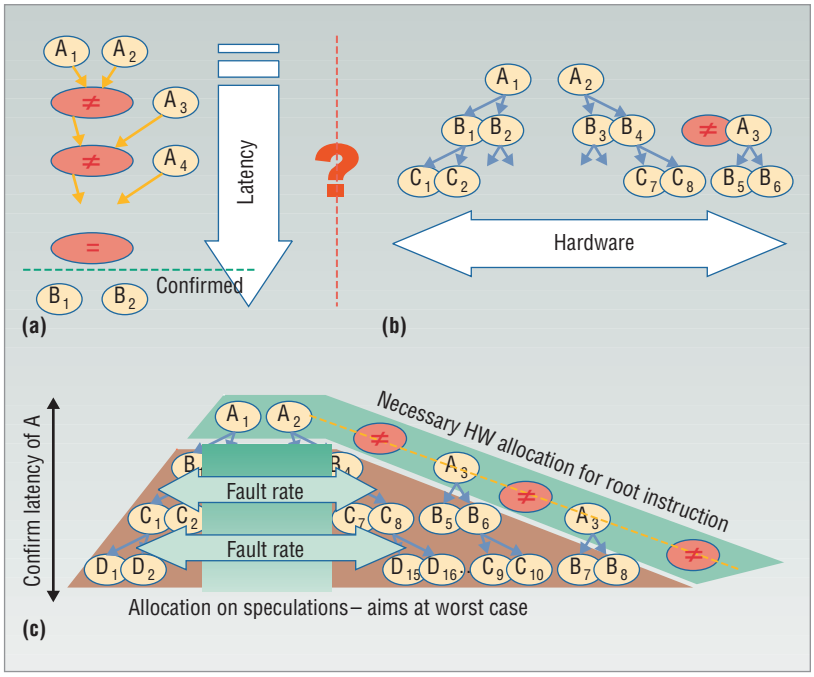


Figure 1. Nanosystem reliability. Temporal (a) and hardware (b) redundancy have traditionally resolved high fault rates. (c) Hybrid architectures necessitate exploration of a speculation-based adaptive mode.

formance in proportion to device density. Performance, in terms of millions of instructions per second per watt, isn't keeping up with the increase in millions of devices per chip. There's also a gap between device density and our ability to design new chips that use every device on the chip and guarantee they're designed correctly.

Power consumption and heat dissipation present additional challenges. The semiconductor industry is investing tremendous effort in finding solutions as we move down this evolutionary path, but it's increasingly difficult to design, fabricate, and test solutions.

Revolutionary path

Knowing the end of Moore's law scaling is in sight for traditional silicon technology, many have embarked on revolutionary nanoelectronics research. Researchers are studying carbon nanotube transistors, carbon nanotube memory devices, molecular electronics, spintronics, quantum-computing devices, magnetic memory devices, and optoelectronics—technologies addressed in the emerging devices section of the *2005 International Technology Roadmap for Semiconductors* (www.itrs.net/Links/2005ITRS/ ERD2005.pdf).

Unfortunately, we won't use many of these devices until it's absolutely necessary to consider a replacement technology. So, how should we use these revolutionary nanoelectronic devices in the interim, especially when these devices haven't demonstrated sufficient reliability and large enough signal-to-noise ratio to guarantee reliable digital computation?

RELIABLE SYSTEMS WITH UNRELIABLE DEVICES

In addition to massive CMOS-scaling efforts, many researchers are pursuing molecular, optical, or quantum devices that they could integrate with CMOS-based digital logic to produce hybrid systems.

While there's no consensus yet about which hybrids will enter production, future nanodevices will certainly have high manufacturing-defect rates. Further, we expect them to operate at reduced noise margins, thereby exposing computation to higher soft-error rates. For non-CMOS nanoscale electronics, operation uncertainties originate in the devices' inherent stochastic switching behavior. Finally, devices will have more process variability—and thereby more nonuniform behavior across a chip—so circuits must be more robust to this process variation to prevent unacceptable yield loss.

Power density and energy cost are the main design bottlenecks for CMOS nanoscale technology. Adding redundancy to increase error resilience eventually increases design complexity, decreasing energy efficiency and compromising density advantages.

Granularity of the fault tolerance is also important. Some redundant techniques improve yield with small cost increases, such as providing spare cache lines to substitute for defective hardware. Others, such as macroredundancy in the form of triplicate voting schemes, are much more expensive.

While nanoscale devices have the advantage of low power, particularly if switching is accomplished without physically moving significant amounts of charge in space, nanoarchitectures will most likely have huge complexities, driven by application needs and the redundancy required to enable fault tolerance. Low-power nanodevices are intrinsically error-prone because thermal fluctuations can easily switch devices across the low-energy barrier separating different logic states.

Temporal and hardware redundancy have traditionally resolved high fault rates, as Figure 1 shows. The unpredictability in confirmation completion and the worst-case hardware overhead require reliable hybrid architectures, necessitating exploration of speculation and adaptivity to ensure correct computation at low hardware and time costs.

High-level issues

Researchers must address several high-level issues in the search for revolutionary architectures for building reliable computers from unreliable devices.

Defect and fault rates. Devices designed in the nanoregime create different problems than those with current VLSI technology. In particular, defect and fault rates, as well as process variability, were never considered "show stoppers." At the nanoscale level, however, high defect rates and variability will be first-order design considerations, not merely add-ons to previously established design objectives. Most effective, novel design approaches must incorporate redundancy at several levels of abstraction.

Synergy between levels. There must be a tight synergy between levels of technology abstraction, which might require passing on more design information from one level of abstraction to the next. Although this might lead to more complex designs, it's required for achieving an appropriate level of reliability.

Well-designed interface. Since this work is interdisciplinary, researchers must clarify interfaces between various levels of abstraction during the tool-development process. Researchers need to understand expectations among different groups before developing a well-defined interface.

Exploring potential. Research in nanoarchitectures for revolutionary computing models will lead to new ways of exploiting the potentials of nanotechnology and nanoelectronics.

Reliability issues will cut across both active device and interconnect design levels and might require regular topologies to enable amortization of reliability overhead. Figure 2 shows the fundamental opportunities and attributes in nanoelectronics that will shape design approaches at various levels of abstraction.

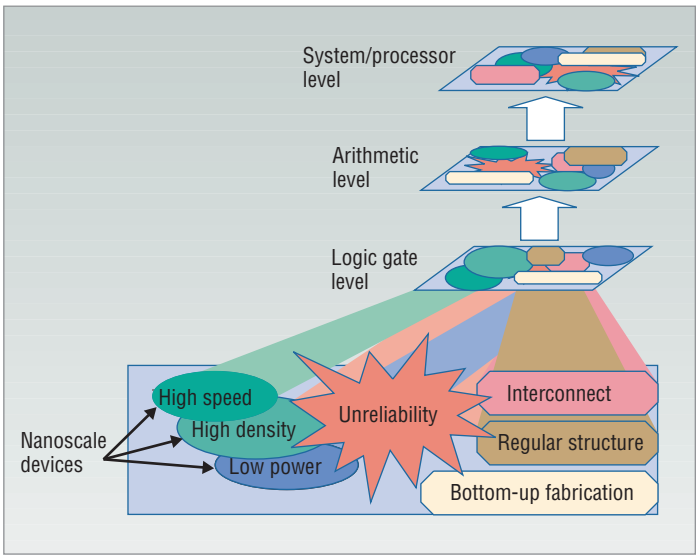


Figure 2. Nanosystem opportunities and attributes. Nanoelectronics will shape design approaches at various levels of abstraction.

Devices and circuits

Computation with nanoscale devices implies computing close to the thermal limit. At this point, computation becomes probabilistic in nature. Along with fault modeling, analysis, and propagation, evaluating these systems' probabilistic behavior requires more theoretical work. Borrowing ideas from stochastic system analysis might be useful here. Researchers need to develop new computational paradigms that take probabilistic implementation into account.

It's still uncertain how much and what kind of noise nanodevices will encounter in real operation. As researchers develop these devices, we'll get a better sense of their behavior. Nevertheless, researchers must base nanoscale architectures on information obtained from modeling and analyzing real nanodevices so that they're making appropriate assumptions about noisy behavior.

Another important issue concerns the degree to which the application itself can tolerate hardware faults, incorrect operations, and so on. Being absolutely fault-free is significantly more expensive than allowing a small number of faults to be visible at the software level.

Circuit designers have relied on different logic styles to obtain area, delay, or power advantages. Due to the nature of molecular-scale circuitry, designers must add a new constraint—reliability—to the optimization equation. We need comparative studies to assess the reliability of these different logic styles and analyze how it might change as devices shrink to the nanoregime.

One approach is to combine reliable elements with unreliable devices, such as hybrid CMOS/nanodevice circuits. Researchers have proposed several such approaches.[1]

Architecture

Examining the need for new architectures requires an understanding of the applications the architecture will execute and evaluating existing architectures' limitations. While the goal is to design reliable, cheaper, and better-performing architectures built from hybrid nanoelectronic circuitry, it's not clear what aspects of current architectures will present the most serious constraints in reaching this goal. For example, how will interconnect and memory bottlenecks limit the ability to handle high fault and defect rates? Are random technology layouts becoming less desirable as a "fabric" for handling defective devices?

Although they've been tried several times over the years, asynchronous self-timed circuits and logic have limited use. Synchronous circuit techniques have always been more cost-effective and have design inertia and tools on their side. But slow signal-propagation times might bring this era to an end in the nanoscale regime.

Researchers must explore asynchronous designs as a means of simplifying global communication and power issues. A globally asynchronous, locally synchronous (GALS) design approach might be the best way to take

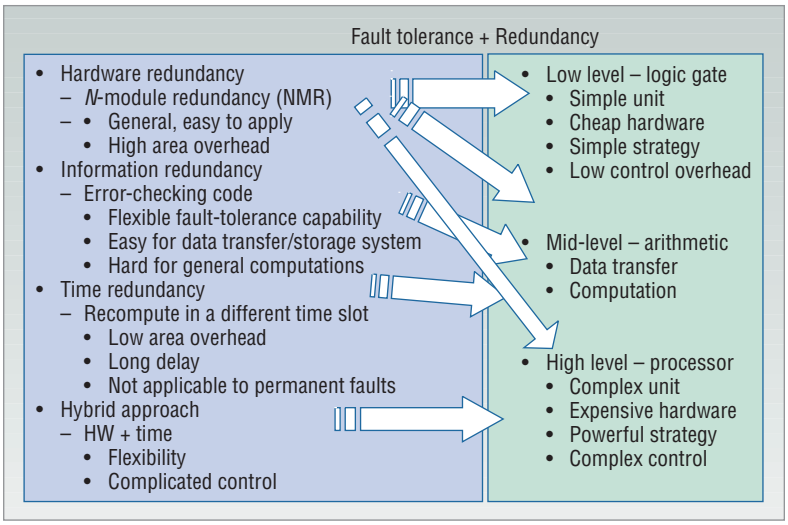


Figure 3. Rough impedance match. Fault-tolerance approaches and applicability of various levels might allow more cost-effective nanosystem design.

advantage of synchrony problems between blocks of nanoscale circuitry. However, GALS and asynchronous designs aren't without their own challenges. Such designs might increase the number of wires, and random noise will be more disruptive.

In addition, such challenges become more involved once we consider faulty connections and devices. The design of fault-tolerant asynchronous hardware is largely unexplored. Ultimately, successful integration of asynchronous designs in future nanoscale architectures will depend on which technologies are viable.

Plausible bottom-up fabrication techniques have demonstrated the feasibility of two-terminal nanodevices for computing applications. Consequently, several approaches to nanoelectronic device architectures have explored ways to leverage two-terminal nanodevices.

While the relatively low functionality of two-terminal devices limits circuit architectures, further research can explore its potential for computing applications. It's possible to build dense regular structures, such as logic grids and memory arrays, which might be the best way to use two-terminal devices when nanodevices first achieve commercial viability.

Reliability theory. Reliability theory has traditionally investigated bounds on system behavior based on simplified assumptions. For example,

- all gates have the same probability of failure,
- only gates fail (and not connections),
- only stuck-at faults are considered, or
- faults aren't state dependent.

Although simplistic, these assumptions have let designers reasonably approximate expected system behavior. On the other hand, these same assumptions might lead to flawed conclusions about expected behavior for systems built from molecular circuits.

We need more realistic characterization of the nature of faults at the molecular scale, as well as an understanding of how faults might manifest themselves in terms of logical and system behavior. We need new fault models for both gates and wires. And researchers should review traditional theoretical results using these new fault models.

We need to identify and optimize algorithms for automating such computations, since they'll be essential in developing fault-tolerant circuits and CAD tools for reliability estimation.

Computational theory. To build effective architectures for reliable computation, we must consider several issues at various levels of abstraction. At the highest levels, we need to explore new models of computation and information representation. Current approaches to data representation might no longer be viable when a system has widespread static and dynamic faults and noise. Consequently, we need to understand issues involved in adding reliability where standard and innovative hybrid techniques might be appropriate. Figure 3 shows a rough impedance match for high fault-rate regimes such as nanoelectronics.

Allowing fault tolerance to operate at different levels of abstraction might facilitate a more cost-effective design. Furthermore, developers can hierarchically implement error detection and correction at various levels of abstraction, as well as represent data using error-correction codes. Hierarchical techniques can also provide avenues for handling fault clustering cost-effectively. We should consider security in parallel with reliability since these two issues might share similar solution spaces.

Fault/defect management. Reliability concerns an entire system, with contributions from all levels. Once researchers develop fault models, they must conduct probabilistic analyses of them. Detecting the faults requires incorporating an effective test-design methodology into the architecture. Another open area of research deals with the testing of fault-tolerant-based circuits.

The reconfiguration or sparing process can be part of defect testing. Handling transient and intermittent faults will require runtime monitoring to detect these soft errors, along with prediction and recovery schemes.

Given the high error rates, it might be more economical to borrow coding techniques from the communications community rather than building in massive redundancy or reconfigurability. However, the design ultimately will need both error-correction codes and redundancy/reconfiguration if minimum area is the goal. Blending the two approaches and achieving the gradual

transition from brute-force redundancy at the very low level to ECC at higher levels of design abstraction will be challenging.

APPLICATIONS: THE DRIVING FORCE

A computer architecture structures a hardware system to provide a reasonably efficient solution to a variety of application problems. As we approach the era of molecular-scale electronics, identifying the applications that these devices will execute becomes a crucial part of the design of nanodevice architectures and circuits.

Today's application-driven architectures are mostly in the multimedia, communication, and pattern-recognition domains. Commercial implementations of these applications might benefit a migration from silicon-based computing systems to hybrid nonsilicon nanoscale technologies. In such cases, it would be possible to integrate nonvolatile, fast nanoscale memory devices with classic nanoscale silicon, or even use devices that can do both processing and storage.

In addition, hybrid architectures will allow integration of sensing and processing functions in ways analogous to biological systems. Living beings can perform complex real-time functions with remarkable ease, unmatched in performance by the most powerful man-made computers. These capabilities are likely due to the seamless integration of sensory, memory, and processing functions in biological systems. Inspired by biology, cellular sensor-processor architectures appear promising for hybrid nanodevices.

Application-driven nanoarchitectures

Nonsilicon nanotechnologies provide opportunities to radically change the architectures employed in next-generation integrated circuits. The availability of very dense conventional silicon technology, along with non-conventional, nanoscale storage or memory technology based on phase-changing materials, makes fascinating hybrid architectures possible.

Integration of logic and memory will allow large-scale array computing with increased local storage. From this perspective, the style of choice seems to be regular or tile-based architectures that rely mostly on local computation and storage while requiring sparse global communication. This is, incidentally, an important operating principle of neural circuits.

While recent research has focused on such massively distributed architectures, the availability of extensive local storage, fully integrated with logic, could offer new capabilities and increased performance for applications currently limited by the logic-memory communication overhead. Relevant work in this area includes the Intelligent RAM project at the University of California, Berkeley.

Having logic and memory in close proximity allows developing improved shared-memory-based architectures, thus possibly enabling heterogeneous communication architectures based on routing data between various tiles of logic or localized shared-memory communication. Figure 4 shows a possible path from program description, via an intermediate format, to a regular, tile-based architecture with localized communication and adaptive reorganization in the face of high defect rates.

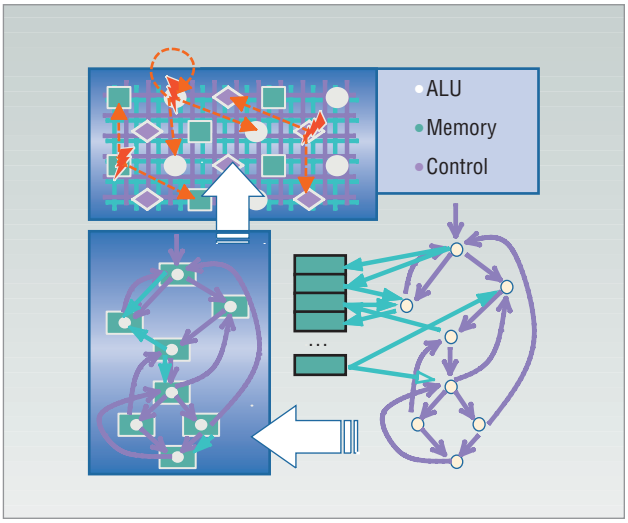


Figure 4. Possible architectural path. The design progresses from program description to regular tile-based processor nanoarchitectures.

In addition to logic and memory, hybrid architectures might allow for integration of sensing with logic and memory. One important class of applications would be vision systems, in which each photo detector would be embedded with its own circuitry in a cellular architecture. Each element in such an architecture would resemble a "neuron" in a retina-like array, where the system performs basic image-processing functions on the incoming image flow.

This merging of sensing and processing offers performance and functional advantages over conventional vision systems, where the functions of sensing and processing are primarily accomplished on separate chips; the required data flow between the focal-plane array and the data processor presents a significant bottleneck for overall system performance.

Similarly, other biologically inspired applications include speech or auditory processing. The human auditory system is ingeniously designed to process sound and speech through the cochlea and the auditory cortex. The human auditory system can recognize speech even if it's garbled, embedded in noise, or mixed with other voices. Here, in addition to dense memories and massively parallel processing, nanoelectronics offers the potential for combining the sensing of sound and the processing of speech into a single computational nanoarchitecture.

Nanostorage

Increasing processor horsepower has traditionally garnered most of the attention in the computer industry. But in many cases, storage technology, both silicon and mechanical/magnetic, has improved faster than processors. We know empirically that an order-of-magnitude quantity change often results in qualitative changes in usage models, so there's ample reason to believe that nanodevices will find new and revolutionary uses because of their increased local nonvolatile storage.

Applications that will benefit from increased storage availability and its integration with logic functions are those whose performance suffers due to storage needs that current technologies can't support. Signal-processing applications rely on complex learning algorithms requiring large storage capabilities, such as handwriting, voice, and speech recognition, and are often based on pattern-matching algorithms or learning algorithms using hidden Markov models requiring massive amounts of memory.

The key challenge is how to build reliable systems from imperfect devices.

Applications involving matching when searching a large collection of data—such as the ones used in biometric identification or gene analysis—will greatly benefit from memory-intensive architectures. Taking this idea even further, separate memory architectures can be built for storing abstract patterns such as text symbols, numbers, or even music patterns, to enable a brain-like memory behavior.

SYSTEMS DEVELOPMENT

Nanoelectronics can potentially put a trillion molecular-scale devices in a square centimeter. New nanoscale electronic devices also offer increased carrier mobility and reduced power dissipation per switching operation. Developing the technological capability to assemble molecular-scale devices is a prerequisite. However, we need much more to allow us to efficiently and cost-effectively assemble a trillion devices per square centimeter with sufficient reliability.

In particular, we must have nanoelectronic circuit fabrics that can harness this enormous number of devices to perform useful computations. Synthesizing and mapping VLSI circuits and systems into this fabric and dynamically reconfiguring the fabric to match the application needs require sophisticated design methodologies, tools, and runtime support.

We need to identify common abstractions and problems underlying nanodevices and fabrics to ensure that design-tool R&D will be independent of the choice of the nanotechnology device or circuit fabric.

Nanodevice and nanocircuit fabric attributes

Nanoscale devices will operate stochastically and require postfabrication correction—that is, programming—to identify and bypass manufacturing defects. They'll only work correctly a portion of the time, and thus need circuitry to support built-in fault detection and correction. In addition, they'll likely have small or no gain and can only support low fan-out, often requiring level restoration.

Nanowires will have a high resistance, be relatively scarce, and typically provide only nearest-neighbor connectivity. These characteristics tend to restrict the ability to distribute a shared signal to multiple destinations at the nanodevice level. Global communication requires wiring technologies and techniques for nano-to-CMOS and CMOS-to-nano.

Key features of nanoscale circuit fabrics include the availability of abundant hardware resources, which facilitate the use of huge on-chip parallelism; fine-grained interleaving of memory and logic devices, which greatly improves the logic-memory bandwidth; categorical dominance of interconnect delays over "gate" delays; and record levels of variability/unreliability of basic devices.

Nanoscale circuit fabrics and architectures therefore demand improved design methodologies and tools to cope with huge numbers of devices with limited connectivity, interconnect parasitics, and unprecedented levels of statistical variability, as well as permanent defects and transient faults. The key challenge is how to build reliable systems from imperfect devices.

Challenges

Projected nanodevice manufacturing processes will be radically different from conventional CMOS processes, at least in the case of bottom-up manufacturing. Because they're still in their infancy, the manufacturing precision is low, resulting in significant statistical variability of each device's key physical, chemical, and electrical properties. These phenomena are exacerbated because the complex and highly controlled patterns that photolithography enables will likely be impractical for nanoelectronic fabrication.

Because of their very small scale, these devices will be susceptible to various kinds of noise, including energy coupling, temperature variations, and single-event upsets. Finally, our limited ability to see what's happening at that scale will make failure analysis extraordinarily difficult.

A nanoscale circuit fabric accommodates terascale devices. Thus, we can classify the key challenges in designing nanoelectronics into several domains: device characterization; simulation, design, and optimization; and system integration at the terascale. The current CAD tools suite can't scale to handle systems with tens of billions of components or more. In addition, they assume lower levels of defects and related faults and hence can't synthesize

nanocircuits with the required levels of fault tolerance, error correction and detection, and diagnosability.

As Figure 5 shows, researchers must synergistically incorporate several factors into traditional methods for reliable design synthesis to construct circuits that map efficiently to underlying nanofabrics and ensure reliable operation despite high levels of fault rates.

Designers must provide certain types of abstractions, such as models of building blocks and reproducibility information, making sure they unify treatment of all layers. From this perspective, we need more studies that address the feasibility of building realistic architectures from nanoscale building blocks and device-specific architectures.

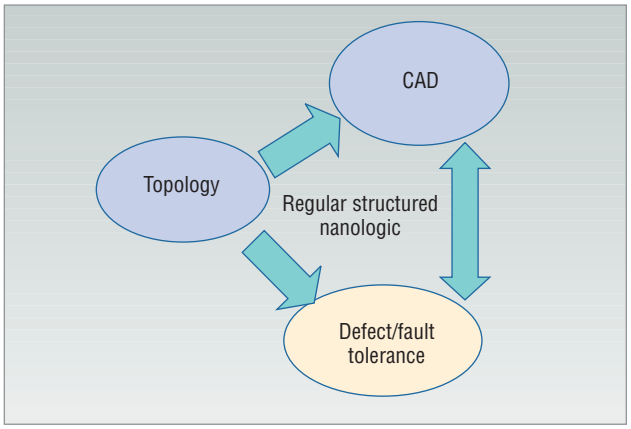


Figure 5. Synergy challenge. A nanosystem design synthesis framework must be synergistic.

Design tools for nanoscale devices and circuit fabrics

The key to effectively using nanoscale devices in electronic circuits and systems is the availability of an accurate yet efficient interface between the physical device technology and the circuit design process.

Models. Models should characterize nanodevice output as a function of the applied input signal. This task might include modeling nanodevices using passive or active circuit elements and developing the corresponding simulation program with integrated circuit emphasis (Spice) on device model parameters. Alternatively, a model might encompass development of an appropriate simulation engine, which captures relevant quantum physical and electrochemical processes in a nanodevice.

The conventional method for modeling devices for use in circuit simulation, such as field-effect transistors, builds on a deep-rooted modeling framework. However, we need to freshly model newly conceived nanodevices. Furthermore, because we don't fully understand the quantum physical characteristics and the underlying physics in these devices, it's difficult to develop compact physics-based models for all the devices that require evaluation at the circuit level.

Logic primitives. Nanodevices tend to use different logic primitives—for example, threshold, majority, and minority functions—compared to CMOS devices, which generally rely on sum-of-product or product-of-sums functions. This means that we need a new suite of logic-synthesis techniques and tools to map arbitrary logic expressions into a netlist of nanotechnology logic primitives. It's conceivable to have a cell-based design methodology for synthesizing terascale integration circuits.

Here, key logic cells are designed based on appropriate nanodevice composition and stored in a cell library for use by the synthesis tools. We should then characterize such nanodevice-based logic cells in terms of their noise margins, propagation delays, and leakage power. What's new here is that we must characterize the nanodevice-based cell libraries in the extended space of delay, power, and reliability, which in turn quantifies effects such as low-defect probability density, transient fault tolerance, and process-manufacturing variability resilience.

Circuit fabrics. Nanodevices will likely be built in regular arrays of nanoblocks, each consisting of many locally connected nanodevices. These nanoblocks will in turn interconnect by using global resources, possibly in the supporting CMOS circuitry. This is an example of a circuit fabric. Crossbar-based circuit fabric is another example. Researchers must characterize and optimize these fabrics in terms of the complexity of nanoblocks, local neighbor connectivity within the blocks, and global interconnection architecture used for interblock communication.

We must be able to compare a new nanodevice-based circuit fabric against existing silicon-based fabrics, hybrid CMOS-nanofabrics and, more generally, any other competing nanocircuit fabric. This requires developing a comprehensive, modular, and flexible evaluation platform to compare and contrast the competing nanocircuit or hybrid CMOS-nanofabric and architectures in terms of the key performance metrics: density, latency, power dissipation, defect/fault/variability resilience, and so forth.

Defects. Researchers must develop tools for locating defects at a sufficient level of granularity on a manufactured nanocircuit fabric to avoid or work around affected nanodevices or nanoblocks. They must also develop an understanding of the various sources of defects and failures in the manufacturing process and develop fault models that accurately capture most behavioral effects these defects produce. Finally, they must develop on-chip test-pattern-generation and response-evaluation hardware to test the various nanoblocks. Some have argued that reconfigurable architectures are naturally defect-tolerant because it is essentially possible to program such architectures to test themselves. In the presence of high defect levels, it isn't clear if this proposition is true.

Design tools. With nanoelectronics enjoying a higher level of device integration than their CMOS counterparts, we expect that it will be necessary to revise or reinvent many existing synthesis, physical design, and verification methodologies and techniques to handle the sheer complexity of nanoscale designs. With the huge number of nanodevices available in a nano-integrated circuit, it's essential to raise the level of design entry and abstraction from register-transfer level to the architecture and system level to manage design complexity and increase design productivity.

We need new design flows and tools to optimize nanoelectronic circuits for performance and yield. More precisely, these tools must be suitable for the nanotechnology fabrication process and address the unique features of nanotechnology devices and fabrics—for example, the multiple-valued logic nature of many such devices, local connectivity within a nanoblock, and so on.

At the same time, the tools ought to optimize the circuit, with error resilience being the key driver more than any other performance metric. It's also imperative to analyze and quantify the key sources of nanodevice and nanowire parameter variability as a result of the nanodevice manufacturing or the nano-CMOS integration processes. Architectures with significant locality will offer distinct advantages.

Finally, it's also useful to have a loosely integrated software-design framework for the various modeling and simulation tools, physical design and synthesis tools, system-level exploration, and architecture-optimization tools so that an expert designer can choose the optimal set to use when mapping a design into a target nanocircuit fabric.

Terascale integration

If history is a guide, there's another problem associated with fault tolerance: defects that stem directly from design errata. As silicon devices have incorporated more transistors, they've also incorporated more design errors. CAD has barely kept pace with the industry's ability to place more active devices on a chip, and verification has not done so at all. Existing and projected design tools and methodologies might not be capable of producing chips approaching a trillion devices. This is a critically important challenge.

Building complex, reliable, and correct circuits and systems from nanocircuit fabrics requires addressing several problems related to integrating so many devices into a single design on the same chip. Design tools are critical to realizing the new architectures that the huge number of switches available at the nanoscale will make possible. These problems include realizing a level of parallelism significantly beyond that used today, integrating memory and logic modules at a low level of granularity, new architectures for heterogeneous integration, and those based on new computation models. Furthermore, researchers must address problems arising from the huge number of defects and failure mechanisms that might exist in a nanodevice-based circuit. Programming nanoblocks will also be challenging.

Design flow. Most important, how can researchers scale current toolsets to handle a trillion switches? Standard design flow and methodologies must change. In particular, a new intermediate representation that could capture a design's high-level characteristics and performance specifications might be useful.

As silicon devices have incorporated more transistors, they've also incorporated more design errors.

There are enough abstractions of key attributes of the target nanodevice and nanocircuit fabric to make the mapping process truthful and reliable. Current work on tools for system-on-chip designs is heading in this direction. Researchers must address design and verification problems having to do with:

- using an inherently randomized structure of nanoblocks with a fabric that either models random defects in a nanoswitch crossbar array or the statistical variations due to the inherent randomness of the chemical self-assembly process in a molehole;
- observability and controllability of such nanoblocks;
- hybrid and hierarchical circuit architectures that harness the potential of nanoblocks and significant investments in the CMOS processes; and
- dynamic reconfiguration of nanoblocks as well as CMOS switching matrices to provide on-the-fly customization of both the underlying nanoblocks and the overall circuit architecture to satisfy the functional or computational performance requirements.

On-chip hardware might support this evolution of the underlying circuit fabric. Reducing the effect of defects requires additional architectural changes. For example, reconfigurable fabrics support defect tolerance in two ways. First, reconfigurability can reduce testing costs. Second, reconfiguring or making adaptations can avoid detected defects. The Teramac work clearly showed that such an approach is viable.[1]

Memory. Emerging types of universal memories might enable new and innovative architectures because they offer new processing/storage tradeoffs compared to existing memories. Properly using the new memories and related system solutions requires system-level synthesis and architecture exploration and optimization methodologies and tools.

Research in the architectures, methods, and tools required to maximally leverage nanoscale devices and terascale capacity can't wait until we've narrowed technology options or achieved volume production. We need research into these areas now because, in addition to enabling systems beyond CMOS, this work is important to benefit current silicon technology approaching the end of the roadmap. Applications, architectures, and models must advance in parallel with efforts in devices and materials.

We must engage the computer architecture community to explore architectures that exploit nanotechnology's benefits and minimize its pitfalls; reawaken parallel computing research with an eye toward mapping more algorithms into parallel forms; consider alternative computation and information representation models; and make fault tolerance, reconfigurability, and power primary issues at the architectural level. This includes interdisciplinary research into applications that can take advantage of nanoscale computing's phenomenal potential density: applications benefiting from massive storage and its integration with logic, signal processing, and other algorithms based on learning.

Researchers also must develop new design paradigms, methods, flows, and tools that leverage devices and interconnects at the terascale, including the ability to deal with the complexities of faulty components and fault-tolerant circuitry. These will include modular, interoperable, and flexible hierarchical modeling and optimization tools, characterizing the nanodevices in the extended space of delay, power, and reliability.

Nanosystems built on this research will encompass multiple levels, ranging from device to computational paradigms and providing solutions in multiple application areas. With the advent of nanoelectronics, we're entering an exciting new phase of computer engineering, where intrepid early explorers will reap significant rewards.

Acknowledgments

This article is the result of a workshop sponsored by the National Science Foundation, grant CCF-0541927, the Semiconductor Research Corp., Portland State University, and the Oregon Nanosciences and Microtechnologies Institute (ONAMI), "Architectures for Silicon Nanoelectronics and Beyond: A Workshop to Chart Research Directions," http://web.cecs.pdx.edu/~strom/nsf_workshop/index.htm.

Reference

1. D.B. Strukov and K.K. Likharev, "Defect-Tolerant Architectures for Nanoelectric Crossbar Memories," *J. Nanoscience and Nanotechnology,* Jan. 2007, pp. 151-167.
2. J.R. Heath et al., "A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology," *Science*, 12 June 1998, pp. 1716-1721.

R. Iris Bahar *is an engineering professor at Brown University. She received a PhD in electrical engineering at the University of Colorado, Boulder. Bahar is a member of the IEEE and the ACM. Contact her at iris_bahar@brown.edu.*

Dan Hammerstrom *is an electrical and computer engineering professor at Portland State University. He also has a joint appointment at Halmstad University in Sweden. Hammerstrom received a PhD in electrical engineering from the University of Illinois at Urbana-Champaign. He is a Senior Member of the IEEE. Contact him at strom@cecs.pdx.edu.*

Justin Harlow *is an instructor and research associate at University of South Florida. He received a master's degree in electrical and computer engineering from Duke University. Harlow is a Senior Member of the IEEE and a member of the AAAS. Contact him at harlow@cse.usf.edu.*

William H. Joyner Jr. *is director of computer-aided design and test at Semiconductor Research Corp., on assignment from IBM. He received a PhD in applied mathematics from Harvard University. Joyner is an IEEE Fellow. Contact him at william.joyner@src.org.*

Clifford Lau *is a research staff member with the Institute for Defense Analyses Information Technology and Systems Division. He received a PhD in electrical engineering and computer science from the University of California, Santa Barbara. Lau is a Fellow of the IEEE and is IEEE-USA Vice President for Technology Policy. Contact him at clau@ida.org.*

Diana Marculescu *is an electrical and computer engineering professor at Carnegie Mellon University. She received a PhD in computer engineering from the University of Southern California. Marculescu is a member of the IEEE and the ACM and chairs ACM's Special Interest Group on Design Automation. Contact her at dianam@ece.cmu.edu.*

Alex Orailoglu *is a computer science and engineering professor at the University of California, San Diego. He received a PhD in computer science from the University of Illinois at Urbana-Champaign. Orailoglu is a member of the IEEE and the ACM. Contact him at alex@cs.ucsd.edu.*

Massoud Pedram *is an electrical engineering professor at the University of Southern California. He received a PhD in electrical engineering and computer science from the University of California, Berkeley. Pedram is an IEEE Fellow and an ACM member. Contact him at pedram@usc.edu.*