The Search for Alternative Computational Paradigms

Naresh R. Shanbhag
University of Illinois at Urbana-Champaign

Subhasish Mitra
Stanford University

Gustavo de Veciana and Michael Orshansky
University of Texas at Austin

Radu Marculescu
Carnegie Mellon University

Jaijeet Roychowdhury
University of Minnesota

Douglas Jones
University of Illinois at Urbana-Champaign

Jan M. Rabaey
University of California, Berkeley

Editor's note:
With statistical behavior replacing deterministic behavior in integrated systems, traditional thinking about computation may no longer apply. This article explores new communication-based models for technologies at the end of, and beyond, the CMOS roadmap.

—William H. Joyner Jr., Semiconductor Research Corp.

EXTREMES OF PROCESS variation, noise, soft errors, and other nonidealities in nanometer process technologies threaten to nullify the intrinsic advantages of scaling that the semiconductor industry has come to expect.[1] Materials and nanodevice research continue to produce candidates for scaled-CMOS and post-silicon-era design. These include devices such as carbon nanotube field-effect transistors (CNFETs), which are known for their excellent switching speeds. An ideal CNFET technology enables the design of digital circuits with a 13× energy-delay product (EDP) advantage and a 5× speed advantage over 32-nm silicon CMOS.[2]

However, these and other post-silicon device candidates suffer from extreme amounts of statistical variation in device behavior, leading to a lack of robustness. Advances in manufacturing technology alone cannot address the robustness problem cost-effectively. Material and device advances must be complemented by innovations in design, test, and verification methodologies. For Moore's law to remain in effect, designers must address energy efficiency, performance, and robustness issues jointly.

Current computational systems are based on the legacy of Von Neumann, Turing, and Boole. This legacy takes a deterministic view of computation and computational substrates, and has served us well for the past five decades. It is not clear that this deterministic computational paradigm is well-suited for the realities of the nanoscale and post-silicon eras, where statistical behavior is the primary attribute of device and circuit fabrics. In fact, assuming independent and localized models of noisy and error-prone logic gates, Von Neumann and Moore and Shannon independently showed that arbitrarily reliable computation is possible with unreliable components.[3,4] Although their approaches predicted the current state of affairs, they had very high overhead, which would nullify any benefits of scaling if these approaches were applied today. Moreover, these approaches were based on overly simple models of the circuit fabric. This raises some important questions: Are there alternative models of computation that can embrace randomness and statistics, treating them as opportunities rather than problems? Can reliable systems be cost-effectively designed using components exhibiting statistical behavior? The semiconductor industry's future could depend on our ability to answer these questions satisfactorily.

In this article, we hypothesize that a networked computational paradigm supported by a device and circuit fabric with an appropriate level of robustness

0740-7475/08/$25.00 © 2008 IEEE Copublished by the IEEE CS and the IEEE CASS

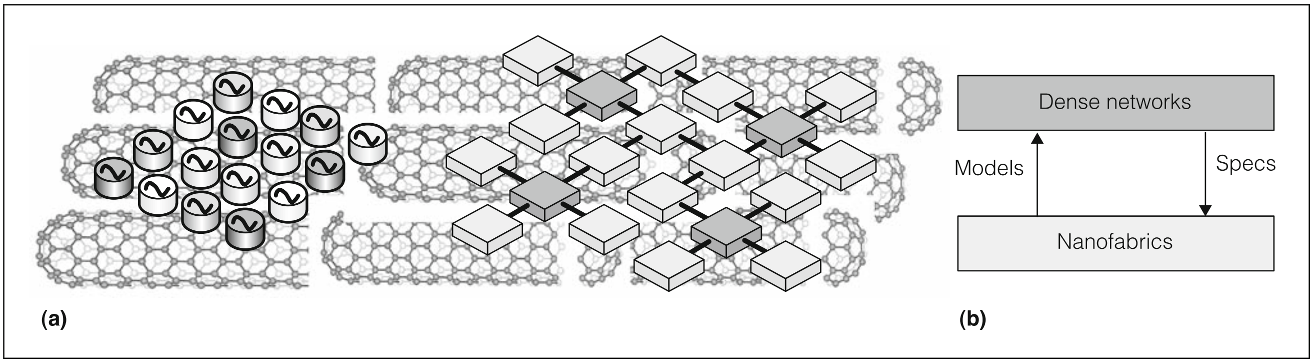

Figure 1. Computation via dense networks: robustness and efficiency via information sharing (a), and relationship between circuit fabric and system design (b). (The round cylinders represent analog nodes; the square boxes are digital nodes.)

can enable the right trade-offs between robustness and energy efficiency in emerging process technologies.

Computation via dense networks

As noted by Shanbhag in 1997,[5] and later by the *International Technology Roadmap for Semiconductors* in 2001, it is undeniable that SoCs in nanoscale process technologies are beginning to resemble communication networks. This resemblance points to the potential of employing communications-inspired, network-based computational models to effectively trade off energy efficiency, performance, and robustness. After all, modern wireless and wire-line networks have been playing the energy (low signal-to-noise ratio) and reliability-robustness (low bit-error rate) game since the publication of Shannon's pioneering work in 1948. Many of the same techniques used in the design of robust communication networks can be used in the design of robust, low-power SoCs.

Networked computation holds particular promise. Networks come in various shapes and sizes, and in domains ranging from electronics to biology to socio-economics. Properly designed networks exhibit a desired globally emergent behavior or functionality as a result of local information exchange. One such behavior is their intrinsic robustness to component (node and link) failure; in other words, the network is a reliable system designed with unreliable components. In contrast, today's SoCs achieve system reliability mainly through component-level reliability. Information exchange for reliability enhancement is nonexistent. This situation gives us the opportunity to investigate alternative computational models that can provide orders-of-magnitude improvement in robustness without compromising energy efficiency or performance.

Figure 1 shows our vision of networked computation: a dense network of many (thousands, perhaps millions) simple, ultra-energy-efficient, and (most likely) highly unreliable computing nodes (analog, digital, or mixed) using emerging nanodevices that collaborate to produce reliable system-level behavior with low-power operation. Robust device and circuit design techniques will focus on bringing the robustness and energy efficiency of the circuit fabric into the ''comfort zone'' of network dynamics. Such an approach is expected to produce an improvement in system-level reliability of several orders of magnitude, similar to the effect that error-control coding has had over noisy communication channels.

The taxonomy of networked computation is extremely diverse. Networks consist of computational nodes and communicating links. Nodes and links can be analog or digital, discrete-time or continuous, and synchronous or asynchronous, and they can incorporate linear or nonlinear transformations. This panoply of possibilities is both exciting and intimidating.

For one thing, relying on networks to achieve computational robustness can lead to increased communication costs. Determining the energy-optimum balance between computation and communication and then operating at this optimum are challenging. This optimum depends on the relative energy efficiency and robustness of the communications and computational fabrics. An ultra-efficient but potentially unreliable communications fabric will be necessary for densely networked computational systems to win out over conventional systems in terms of

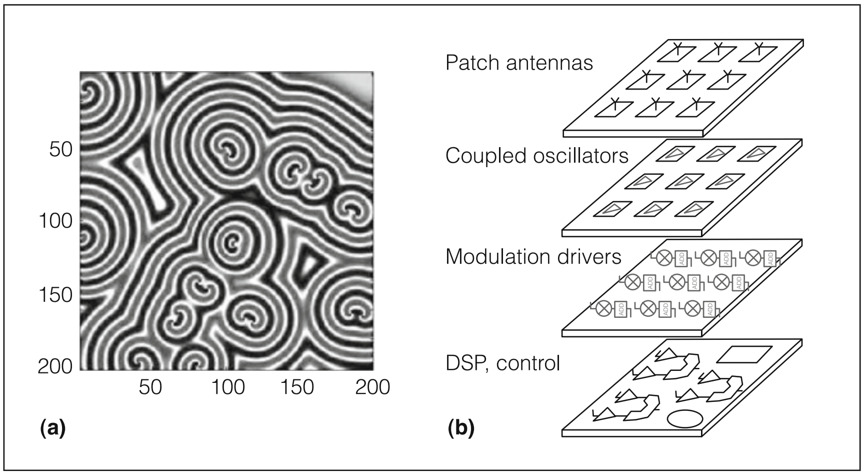

Figure 2. Coupled oscillator network: spatial phase distribution (phase space) of a 200 × 200 oscillator array obtained after 100 oscillation cycles (a), and an application of the network in an RF front-end design (b).

energy efficiency and system-level robustness. The diversity of networks necessitates a corresponding diversity in the mathematical techniques used for modeling, analyzing, predicting, and optimizing network behavior. Tools that could provide strong theoretical underpinnings for the dense network paradigm include nonlinear partial-differential equations, statistical estimation and detection, and inference techniques. Formulating meaningful test and verification metrics and algorithms is another important aspect of this problem. Finally, designers will need a good understanding and appreciation of nanodevice and nanocircuit fabric properties embodied in simple but accurate models and metrics that capture the statistical nature of delay, power, robustness, and behavior.

The following are some of the promising dense-network-based computational models being explored today.

Coupled-oscillator network

The well-known phenomenon of injection locking occurs when two or more oscillators in close proximity start to frequency-lock (operate at the same frequency) and then eventually phase-lock due to coupling. The coupling occurs because of the presence of unintended coupling mechanisms in the supply grid and substrate. Scientists have proposed such mechanisms, for example, to explain the formation of biological patterns such as animal fur patterns, human fingerprints, heart muscle contraction patterns, and others.[6]

Figure 2a shows the phase space (a 2D map of the phase relationship) of a 200 × 200 oscillator array, obtained through simulation of the oscillator array dynamics. The image indicates cooperative behavior among the oscillators as a result of the coupling strength exceeding a threshold. The specific phase pattern is also a function of the initial phase relationship between oscillator elements. A coupled oscillator array's phase space is very robust to large variations in component parameters and coupling strengths and thus represents a potentially attractive computational model in nanoscale technologies.

Because of its analog nature, such a network's most obvious and immediate application is in the design of robust, low-power RF front ends for wireless communications. As Figure 2b shows, the proposed RF front end's architecture is conceptually organized as a four-layer design. The topmost layer is an array of patch antennas organized in a phased-array pattern for beam forming. The second layer consists of a 2D robust network of coupled oscillators that generate appropriate phase patterns to drive the patch antennas. Modulation/demodulation drivers in the third layer drive the oscillator network. The lowest layer, composed of DSPs and logic, produces and receives information symbols for the front end.

The coupled-oscillator framework includes several interesting research topics. One is fast simulation techniques for predicting the behavior of such complex nonlinearly coupled networks. Another topic is driving these networks into predictable states from known initial states. Characterizing the network's robustness to parameter variations in the oscillators and the coupling mechanisms is yet another interesting topic. Low-quality oscillators in nanoscale processes and new applications of such an array are exciting avenues for further exploration.

Stochastic sensor NoC

Recently, researchers have studied sensor networks extensively. These networks are robust to the loss of a few nodes. A stochastic sensor network on a chip seeks to exploit the robustness of sensor networks to enhance on-chip computation. Figure 3a illustrates the SSNoC concept.[7] Traditionally, a main computational block generates a desired output $y[n]$. Because of its centralized nature, this form of computation is vulnerable to localized sources of nonidealities, such as particle hits, hot spots, and across-die process variations, and hence can result in hardware errors.

In an SSNoC, the main or original computation is decomposed into M lower-complexity sensors with complexity ratio R, where R is the complexity ratio of one sensor to that of the main computation. The sensor outputs $y_i[n]$ ($i = 1, \ldots, M$) are statistically similar; that is, for $1 \leq i \leq M$,

$$\begin{aligned} y_i[n] &= y[n] + \eta_i[n] \\ E\{y_i[n]\} &= y[n] \\ E\{\eta_i[n]\} &= 0 \end{aligned}$$

The gray shading around some of the black dots in Figure 3a represents the fact that instantaneous sensor outputs $y_i[n]$ might not equal the correct output. Thus, an SSNoC is characterized by the two key parameters, complexity ratio R and decomposition factor M (the number of sensors), as well as the fusion block functionality and implementation. We can make several observations:

- If $R = 1$ and the fusion block is a majority voter, the SSNoC becomes equivalent to N-modular redundancy (NMR)—that is, NMR falls out as a special case of SSNoC.
- If $R = 1/M$, the only hardware overhead in SSNoC is the fusion block.
- R and M can be chosen more or less independently, resulting in a family of SSNoC architectures.
- The sensor error $\eta_i[n]$ consists of two error sources: estimation errors ($\eta_{\text{e}\_i}[n]$) from the use of low-complexity sensors, and hardware errors ($\eta_{\text{h}\_i}[n]$) due to the nonidealities in the circuit and process; that is, $\eta_i[n] = \eta_{\text{e}\_i}[n] + \eta_{\text{h}\_i}[n]$.

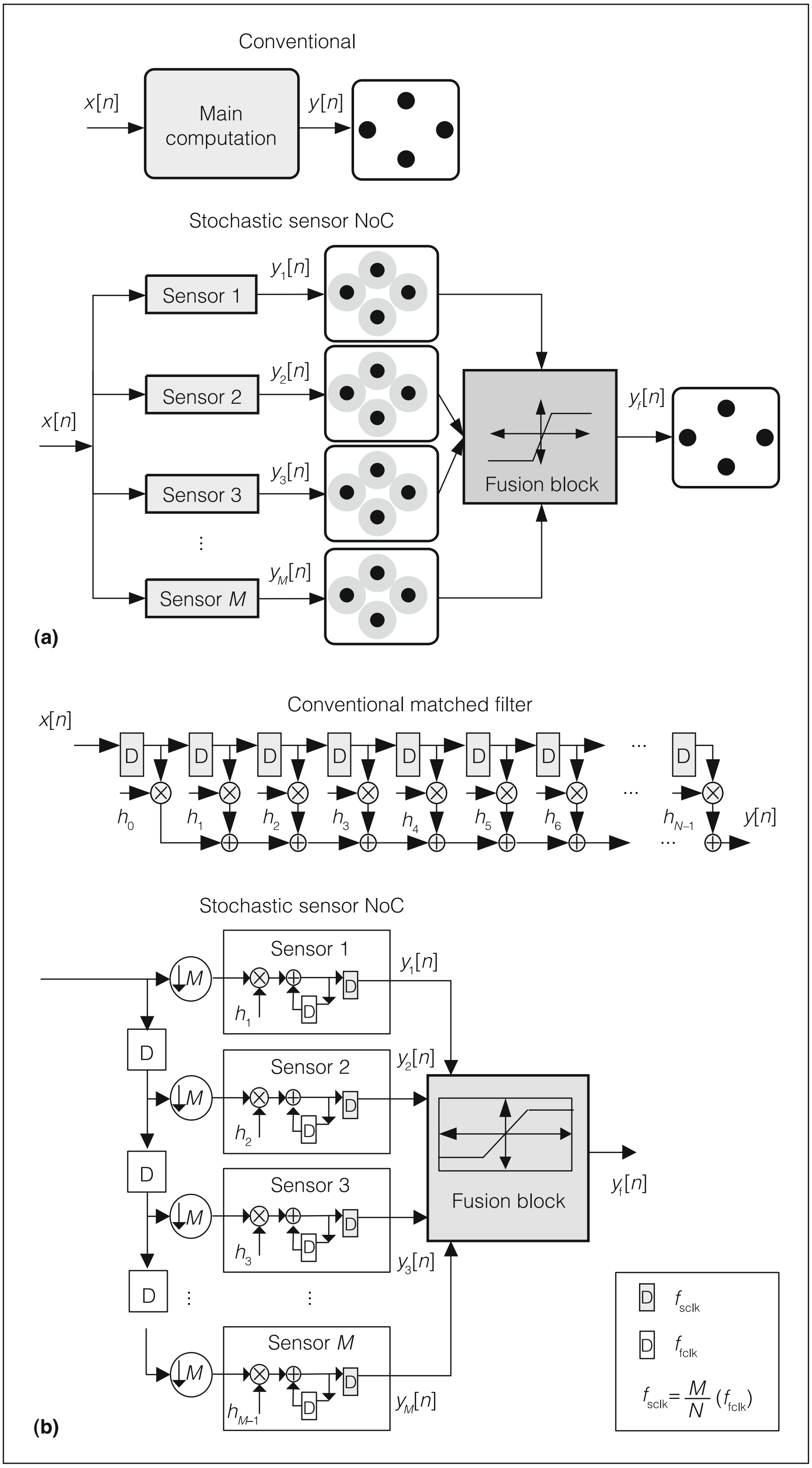

Figure 3. Stochastic sensor network on a chip (SSNoC): statistically similar decomposition (a), and application to pseudonoise (PN) code acquisition for wireless communications (b).

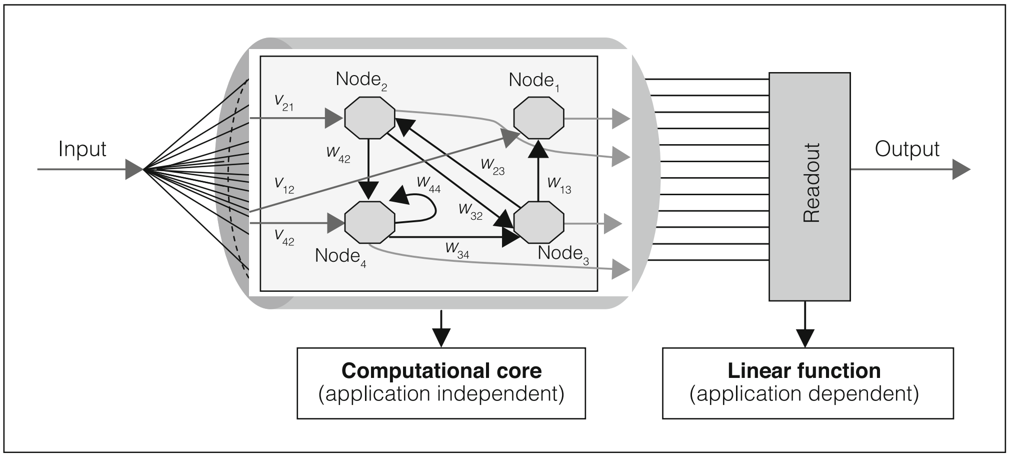

Figure 4. Perturbation-based computing. The computational core is a high-dimensional dynamic system. The readout is a linear function of the states of the computational core.

- As R decreases, $\eta_{\mathrm{e}\_i}[n]$ increases and $\eta_{\mathrm{h}\_i}[n]$ usually decreases. SSNoC complexity is $C_{\mathrm{SSNoC}} = RMC_{\mathrm{orig}} + C_{\mathrm{fusion}}$, where C_{orig} and C_{fusion} are the complexity of the original or main computation and the fusion block, respectively.

Figure 3b illustrates an application of the SSNoC concept to a pseudonoise (PN) code-acquisition matched filter typically employed in code division multiple-access (CDMA) wireless systems. The matched filter correlates received samples with a known PN sequence. It flags a detection event if the correlation is greater than a specified threshold. A polyphase (M-phase) decomposition of the matched filter generates an SSNoC. Each sensor in the SSNoC correlates subsampled versions of the input and PN sequence. A simple addition of the M sensor outputs in the fusion block gives the same result as the conventional architecture.

The key to designing an SSNoC is determining a simple yet effective, potentially nonlinear processing of sensor outputs to generate the final SSNoC output $y_f[n]$, which is statistically close to correct output $y[n]$. Thus, a key question is how to combine sensor outputs $y_i[n]$ to achieve a robust estimate of $y[n]$. Sensor output error $\eta_i[n]$ can be modeled as a random variable drawn from a Gaussian distribution with probability $(1 - \varepsilon)$ and some unknown distribution with probability ε for some $0 < \varepsilon < 1$; that is, an ε-contaminated distribution $f(x)$. The Gaussian distribution represents the estimation error $\eta_{\mathrm{e}\_i}[n]$, and the unknown distribution represents hardware errors due to nanometer nonidealities. Huber identifies the class of estimators known as M-estimators that can be used to compute a theoretically optimum robust estimate.[8]

For $M = 8$, the SSNoC-based PN code acquisition system in Figure 3b can improve detection probability P_{det} by close to three orders of magnitude under the same process and voltage conditions. It can also reduce variation in P_{det} by two orders of magnitude and save 31% power over a conventional architecture. These results clearly show the promise of an SSNoC-based computational paradigm. However, numerous interesting problems remain. These include investigating statistically similar decompositions for media kernels and other generalized computations, obtaining improved fusion algorithms, exploring SSNoCs based on intersensor information exchange, and characterizing power and performance trends over various technology nodes, including post-silicon devices.

Perturbation-based computing

Perturbation-based computing performs real-time computational tasks on time-varying input streams, using the transient perturbations they induce on a high-dimensional dynamic system.[9] Specifically, the user excites a high-dimensional dynamic system such as a complex recurrent network with time-varying stimuli to be processed. This creates a rich pool of dynamics containing several nonlinear combinations of components of the (past) stream. With the nonlinear projections of the original input stream to a high-dimensional space, the user can then train simple, or memoryless, linear readout elements, which produce the desired (task-specific) output stream in real time. Figure 4 shows the elements of this computational framework.

In principle, such systems have universal computational power on time-varying inputs because they can approximate any time-invariant I/O map with fading memory to any degree of precision.[9] This concept has been proposed as a plausible model for the operating principles of biological neural networks. However, information-encoding and computational methods for neural systems are largely open problems. Moreover, this approach diverges from standard computational models underlying state-of-the-art computers and processing engines that require the system to maintain or converge to stable internal states or attractors. This is true of Turing machines, finite-state machines and automata, and attractor neural networks.

Potentially, perturbation-based computing can overcome its challenges by synergistically addressing the two sides of the complex system design equation: technology and applications. Three features of perturbation-based computing address technology issues. First, it is inherently resilient to noise and, therefore, to soft faults and performance variability or fluctuations. Second, in principle, the computational core, a complex dynamic network, can be randomly assembled, taking full advantage of the formidable densities achieved by nanoelectronics technology, while relaxing manufacturing precision and stability requirements. This circumvents the need for design and fabrication of complex structured circuits. As a result, we have the third feature, an inherent tolerance to manufacturing defects or hard faults; these simply become part of the computational core's (desirable) structural randomness. Clearly, these three features make perturbation-based computing almost ideal for technologies exhibiting high defect densities and susceptibility to the soft faults projected for emerging nanoscale processes.

Three additional characteristics make perturbation-based computing suitable for addressing application needs in next-generation IT systems. First, it has an inherent high degree of parallelism. Second, it reduces the required design effort because the same computational core can be used for myriad tasks. Finally, it has the potential of delivering very high energy-delay efficiency. In the future, it's conceivable that very inexpensive perturbation-based computing cores will target specific applications that perform dedicated complex tasks such as voice recognition, video motion compensation, sensing and surveillance, and control.

Stochastic communication

SoCs have evolved into complex networks of dozens or even hundreds of predesigned IP cores assembled to provide complex functionality. These are distributed, nanoscale, multicore systems in which concurrency and communication play central roles. Such systems must be supported by a high-throughput, robust, and energy-efficient communication fabric to reap the benefits of distributed or networked computational models. Stochastic forms of on-chip communication can prove particularly effective.

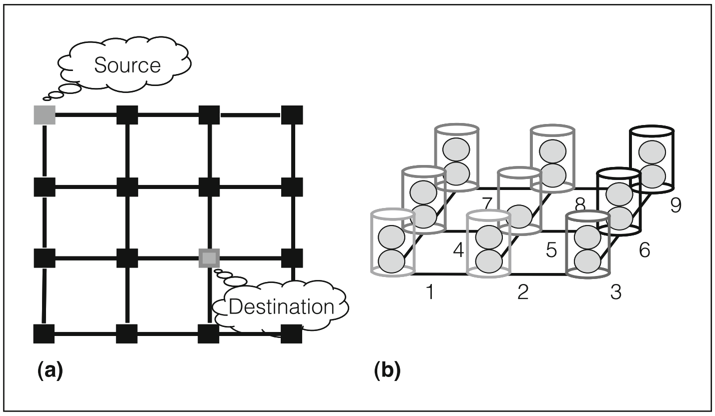

Figure 5. Stochastic communications: biologically inspired network (a), and statistical-physics-inspired virtual random growing network (b).

Stochastic communication uses a probabilistic broadcast scheme for internode communication—a scheme similar to randomized gossip protocols in distributed databases or sensor networks.[10] We can illustrate stochastic communication with a regular array of tiles wrapped in a unified communication interface with input and output buffers, as in NoCs. A node in a tile transmits a packet to a randomly selected subset of its neighbors, which then select only those messages that have their own IDs as the destination. Error detection determines whether the received data is correct. There are no retransmission requests, because multiple copies of the data diffuse through the network, thus making the likelihood very high that a correct version will arrive at its destination. This probabilistic approach is robust to link and node failures but needs additional communication overhead to support redundant communications.

The fundamental equations governing data transmission in stochastic communication have their origins in biology and statistical physics. Figure 5a shows a biologically inspired version based on the dynamics of epidemics spreading in natural populations. Nodes are classified as spreaders (nodes that disseminate packets), ignorants (nodes outside the communication area), and stiflers (nodes that terminate packet dissemination). We can describe the system formally with a probabilistic framework that explicitly captures the interaction between the three types of nodes.

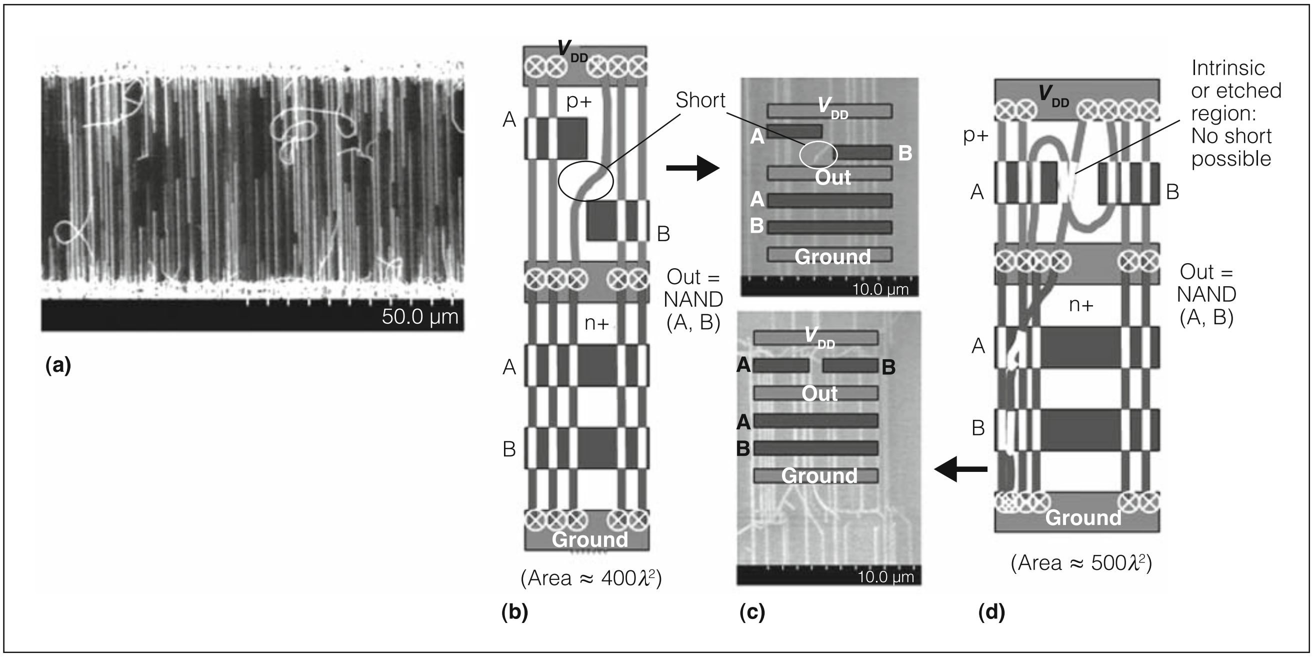

Figure 6. Carbon nanotube (CNT) issues: largely aligned CNTs with misaligned CNTs (a), layout of misaligned-CNT-vulnerable NAND gate (b), scanning electron microscope (SEM) image of CNFET overlaid with gates (c), and misaligned-CNT-immune CNFET-based NAND gate (d).

Figure 5b shows a statistical-physics-oriented approach that views the network of nodes as an interconnected ensemble of buckets known as a virtual random growing network (VRGN). Each bucket contains balls representing packets at a particular point in time. Each bucket or node is assigned an energy level. The VRGN captures the movement of packets as particle transitions among different energy levels in a thermodynamic gas. Mean field analysis shows that the buffer occupancy, as well as the in and out degree of the VRGN nodes, follows a power law. This observation can lead to a fundamentally new approach for on-chip buffer sizing that will provide increased performance and robustness at far lower area overhead and power cost.

Stochastic communication is a major departure from classical (deterministic) bus-based communication. In practice, designers could combine stochastic and conventional communication structures to extract their best features. For example, stochastically communicating islands could be connected to a traditional bus or assembled in a hierarchy, depending on application requirements.

Nanofabric technologies

Exploration of alternative computational models must go hand-in-hand with an understanding of the properties of promising nanoscale device and circuit fabrics. As mentioned earlier, 1D nanodevices such as CNFETs are promising post- or extended-silicon devices. However, much remains to be done to harness the science into practical design techniques competitive with CMOS. These techniques can help bring the level of robustness into the comfort zone of the networked computational models described here.

Transforming materials and device-level innovations into practical technologies for gigascale digital ICs requires overcoming some fundamental barriers: misaligned carbon nanotubes (CNTs); metallic CNTs in CNFETs; and device integration with high CNT density. Of these, the third is a processing challenge that device and materials researchers are addressing. Overcoming the first two requires radical approaches to CNFET-based digital design.

As Figure 6a shows, chemical self-assembly that produces mostly aligned CNTs can partially alleviate the problem of misaligned CNTs. As Figure 6b and Figure 6c show, the remnants of misaligned CNTs cause shorts and incorrect logic functions.[11] Similarly, no known CNT growth technique guarantees the total absence of metallic CNTs. Semiconducting CNTs are required for CNFETs; metallic CNTs create source-drain shorts resulting in excessive leakage and severely degraded noise margins. Post-growth metallic CNT removal techniques, although promising, cannot guarantee removal of all metallic CNTs. To be feasible, defect and fault tolerance techniques must be low cost. Furthermore, if possible, the design of robust CNT circuit fabrics should impose minimal changes on design methodologies.

An imperfection-immune design paradigm can help overcome these barriers. For example, the technique for designing misaligned-CNT-immune circuits guarantees correct logic functions even in the presence of several misaligned CNTs.[11] Figure 6d shows a misaligned-CNT-immune NAND gate. Its layout guarantees that any CNT not passing under any gate in the pull-up circuit has at least one region either intrinsic (undoped) or etched out. This constraint guarantees that such CNTs cannot cause shorts or incorrect functions. Furthermore, this technique can be generalized and automated for any arbitrary logic function; and, for many misaligned CNTs, correctness can be formally proved. The technique is compatible with standard design flows and has low (10% to 15%) area, delay, and power penalties at the cell level. Metallic CNTs require joint optimization of circuit design techniques and processing techniques such as metallic-CNT removal.

We can also use logic-level techniques to control the robustness of these post-silicon circuit fabrics. Communications-inspired techniques such as coding can enhance the robustness of circuit fabrics, but the transformative nature of logic operations makes extending coding for circuits very difficult.

Figure 7 shows an effective functional-coding approach that provides low-level protection of individual Boolean functions.[12] This approach exploits the functional structure of Boolean functions combined with a lookup table (LUT)-based implementation (for example, a ROM) of the Boolean function to produce better codes. Specifically, Boolean functions, and hence their LUT implementations, contain several don't-care values. An efficient algorithm based on a conjunctive normal form satisfiability (CNF-SAT) formulation can exploit these don't-care values to reduce the number of coded bits and hence the number of redundant columns.

Simulations showed that the functional-coding strategy increases the yield from 10% to 90% at a 1% defect probability. By utilizing don't-care values, the technique reduced the number of redundant columns from 3 to 2 for 80% of the LUTs, corresponding to an average area savings of 23%. These results are promising because they show that the coding can greatly increase the yields of circuit blocks even in the presence of extremely high defect densities. This approach enables the use of heterogeneous CMOS-CNT fabrics in which the decoders consist of reliable but inefficient CMOS devices, and the rest of the components consist of unreliable but efficient CNT devices.

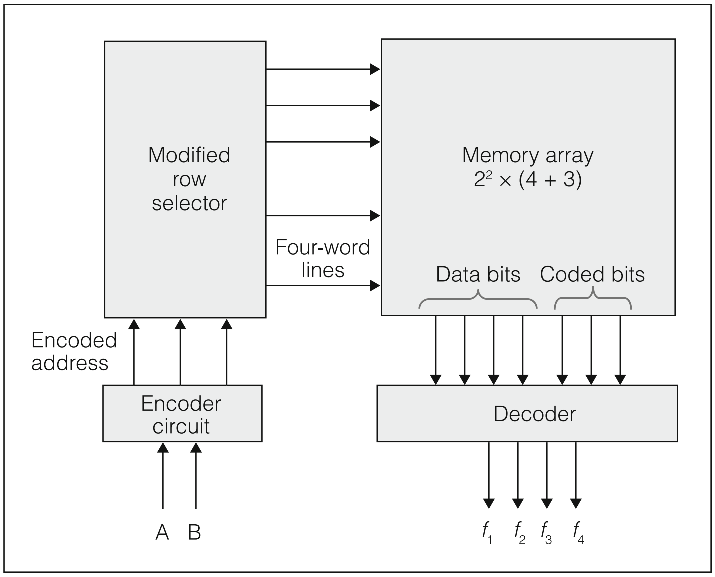

Figure 7. Functional coding with a lookup table.

THE REALITIES OF THE nanoscale regime are forcing the semiconductor and EDA industries to make a transition from the deterministic to the statistical realm. This transition could require a complete overhaul of how we view on-chip computation, communication, and storage. Successful solutions will necessarily involve probabilistic and statistical techniques for analysis and design at the circuit, logic, architectural, and system levels. By its nature, the problem of designing robust, energy-efficient, high-performance systems in nanoscale process technologies is multidimensional; hence, it is difficult but solvable. Collaboration between communication and information theorists, biologists, architects, circuit designers, CAD researchers, and device physicists is essential. A new generation of engineers must be comfortable with the statistical mode of thinking. This transition, though painful, is also exciting and necessary for the continued progress of post-silicon-era design. ■

Acknowledgments

We acknowledge the support of the Gigascale Systems Research Center (GSRC).

References

1. *International Technology Roadmap for Semiconductors*, http://www.itrs.net/Links/2006Update/2006UpdateFinal.htm.
2. J. Deng et al., ''Carbon Nanotube Transistor Circuits: Circuit-Level Performance Benchmarking and Design Options for Living with Imperfections,'' *Proc. Int'l. Solid-State Circuits Conf.* (ISSCC 07), IEEE Press, 2007, pp. 570-588.
3. J. Von Neumann, ''Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components,'' *Automata Studies*, C.E. Shannon, and J. McCarthy, Eds., Princeton Univ. Press, 1956, pp. 43-98.
4. E.F. Moore and C.E. Shannon, ''Reliable Circuits Using Less Reliable Relays,'' *J. Franklin Institute*, vol. 262, Sept. 1956, pp. 191-208, 281-297.
5. N.R. Shanbhag, ''A Mathematical Basis for Power-Reduction in Digital VLSI Systems,'' *IEEE Trans. Circuits and Systems,* part II, vol. 44, no. 11, Nov. 1997, pp. 935-951.
6. A.J. Koch and H. Meinhardt, ''Biological Pattern Formation: From Basic Mechanisms to Complex Structures,'' *Reviews of Modern Physics*, vol. 66, no. 4, Oct. 1994, pp. 1481-1507.
7. G. Varatkar et al., ''Sensor Network-on-Chip,'' *Proc. IEEE Int'l Symp. System-on-Chip*, IEEE Press, 2007.
8. P. Huber, *Robust Statistics*, John Wiley & Sons, 1981.
9. W. Maass, T. Natschlager, and H. Markram, ''Real-Time Computing without Stable States: A New Framework for Neural Computation Based on Perturbations,'' *Neural Computation*, vol. 14, no. 11, Nov. 2002, pp. 2531-2560.
10. P. Bogdan, T. Dumitras, and R. Marculescu, ''Stochastic Communication: A New Paradigm for Fault-Tolerant Networks-on-Chip,'' *VLSI Design,* vol. 2007, article 95348 (17 pp.).
11. N. Patil et al., ''Automated Design of Misaligned-Carbon Nanotube-Immune Circuits,'' *Proc. 44th Design Automation Conf.* (DAC 07), ACM Press, 2007, pp. 958-961.
12. A.K. Singh et al., ''A Heterogeneous CMOS-CNT Architecture Utilizing Novel Coding of Boolean Functions,'' *Proc. IEEE Int'l Symp. Nanoscale Architectures*, IEEE Press, 2007, pp. 15-20.

Naresh R. Shanbhag is a professor in the Electrical and Computer Engineering Department at the University of Illinois at Urbana-Champaign and a research professor in the university's Coordinated Science Laboratory. His research interests include communications IC design, VLSI architectures for DSP, and robust and low-power IC design. He has a PhD in electrical engineering from the University of Minnesota. He is a Fellow of the IEEE.

Subhasish Mitra is an assistant professor in the Departments of Electrical Engineering and Computer Science at Stanford University, where he leads the Stanford Robust Systems Group. His research interests include robust system design, VLSI design and test, computer architecture, and design for emerging nanotechnologies. He has a PhD in electrical engineering from Stanford University.

Gustavo de Veciana is a professor of electrical and computer engineering at the University of Texas at Austin. His research interests include analysis and design of communication networks, and architectures and algorithms for designing reliable computing and network systems. He has a PhD in electrical engineering from the University of California, Berkeley. He is a senior member of the IEEE.

Michael Orshansky is an assistant professor of electrical and computer engineering at the University of Texas at Austin. His research interests include design optimization for robustness and manufacturability, statistical timing analysis, and design in fabrics with extreme defect densities. He has a PhD in electrical engineering from the University of California, Berkeley. He is a member of the IEEE.

Radu Marculescu is a professor of electrical and computer engineering at Carnegie Mellon University. His research interests include developing system-level design methodologies and tools for SoC design, on-chip networks, and ambient intelligence. He has a PhD in electrical engineering from the University of Southern California. He is a senior member of the IEEE.

Jaijeet Roychowdhury is an associate professor of electrical and computer engineering at the University of Minnesota. His research interests include all quantitative and numerical aspects of design and analysis of engineering and physical systems. He has a PhD in electrical engineering from the University of California, Berkeley.

Douglas Jones is a professor of electrical and computer engineering at the University of Illinois at Urbana-Champaign. His research interests include DSP and communications—nonstationary signal analysis, adaptive processing, multisensor data processing, and various applications. He has a PhD in electrical engineering from Rice University. He is a Fellow of the IEEE.

The biography of **Jan M. Rabaey** is on p. 302 of this issue.

■ Direct questions and comments about this article to Naresh Shanbhag, Coordinated Science Laboratory, 1308 W. Main St., University of Illinois at Urbana-Champaign, Urbana, IL 61801; shanbhag@illinois.edu.

For further information about this or any other computing topic, please visit our Digital Library at http://www.computer.org/csdl.

Variability and New Design Paradigms

Leon Stok, IBM

In the late- and post-silicon eras, variation of all nanometer processes will continue to increase significantly. All electrical parameters—such as timing, power, and noise—are already becoming increasingly more affected by these variations. In addition, several effects that can render circuits not only variable, but even unreliable, are becoming more costly to avoid.

Up until the 65-nm technology generation, most variability had been hidden from the designers and had been dealt with in the process of characterizing the technology and generating the device models. Analog and memory designers would run statistical simulations of their designs, but most digital designers were shielded from this. This practice no longer holds for current technology nodes. The extraordinary amount of guard-banding required to sustain this model renders new technologies ineffective.

The industry is gradually addressing this situation and exposing more variability information to the designer. Design tools will attempt to make this information as accurate and actionable as possible, and designers will react with new designs that are more robust and less sensitive to the variations that the analysis tools tell them about. The first statistical analysis tools are being successfully deployed to more designers, and semiconductor fabs are becoming increasingly sophisticated in providing statistical models for their technologies.

It is refreshing to see a university research program like the Gigascale Systems Research Center (GSRC) start at the other end of the spectrum—declaring that the deterministic era will be over for most on-chip applications and that alternatives must be found. The search for these alternative computational models cannot start early enough. For, even if they are found, the paradigm shift to use them effectively could take a long time to implement.

A critical point in the search for these new paradigms is their effect on the power, performance, and cost of design. If too much overhead must be added such that a new design, even with the advantages of the new technology node, is not competitive along these dimensions, there will be no incentive to move to a new, and therefore risky, paradigm. To be viable, new paradigms will need to give at least a 10× advantage over existing paradigms in the current technology node.

I am looking forward to the time when the two will meet: when more revolutionary design techniques will find their way into practical designs, and when one of the computational paradigms will suddenly be needed to cope with an unexpected surge in variability or reliability in a particular design or technology. This moment might be closer than we think.

Leon Stok *is director of electronic design automation for IBM. Contact him at leonstok@us.ibm.com.*

0740-7475/08/$25.00 © 2008 IEEE