ElastIC: An Adaptive Self-Healing Architecture for Unpredictable Silicon

Dennis Sylvester, David Blaauw, and Eric Karl
University of Michigan, Ann Arbor

Editor's note:
The world is coming to its end—well, maybe just the CMOS world. ElastIC must deal with extremes: a multiple-core processor subjected to huge process variations, transistor degradations at varying rates, and device failures. How can we keep the system running? Read on, and find out.
—T.M. Mak, Intel

WITH CONTINUED TECHNOLOGY scaling, silicon is becoming increasingly less predictable. Recent years have brought an acceleration of wear-out mechanisms, such as oxide breakdown and negative bias temperature instability (NBTI), which occur over a part's lifetime. Researchers expect manufacturing device failure rates to increase significantly with decreases in device sizes, possibly reaching one in thousands or even hundreds of devices.[1,2] Process variations will increase significantly in future technologies because fundamental laws of physics drive certain parametric variations, such as random dopant fluctuation (RDF) and line edge roughness, making their increased contribution to variability almost inevitable. In fact, Chen recently described extreme variations due to RDF and line edge roughness as fundamental barriers to controlling device parameters.[3]

The combination of wear-out mechanisms, RDF, and line edge roughness leads to an unpredictable silicon fabric that poses a major obstacle to reliable computing in future technologies. Evidence of this trend's importance is the occurrence of the word "variability" a staggering 73 times in the "Design" chapter of the 2005 *International Technology Roadmap for Semiconductors* (http://public.itrs.net). Researchers have proposed a range of circuit techniques and design methodologies to combat process variability effects (see the "Related work" sidebar), but these solutions are largely ad hoc and sparsely used in industry.

In this article, we present a broad vision of a new cohesive architecture, ElastIC, which can provide a pathway to successful design in unpredictable silicon. ElastIC is based on aggressive runtime self-diagnosis, adaptivity, and self-healing. It incorporates several novel concepts in these areas and brings together research efforts from the device, circuit, testing, and microarchitecture domains. Architectures like ElastIC will become vital in extremely scaled CMOS technologies (such as 22 nm); ideally, they will target applications such as multimedia, Web services, and transaction processing.

Overview of ElastIC architecture

As Figure 1 shows, the ElastIC architecture integrates circuit-, microarchitecture-, and system-level techniques. It consists of four key components. The first component comprises tens to hundreds of small, simple, extremely adaptive processing elements. These PEs contain reliability, performance, and power monitors; BIST components;[4] and associated access mechanisms.[5] Moreover, they are individually tunable through voltage or frequency scaling; bias levels; and dynamic, soft cycle boundaries.

The second component is a central diagnostic and adaptivity processing (DAP) unit. During runtime, the DAP unit takes each PE offline in turn to perform detailed diagnostics of parametric variations and wear-out using ATPG coupled with in situ reliability monitors in each core. The DAP unit tracks the degradation of individual devices or small device clusters caused by wear-out

0740-7475/06/$20.00 © 2006 IEEE Copublished by the IEEE CS and the IEEE CASS

Related work

Researchers have proposed various approaches to statistical static timing analysis (SSTA) and optimization, with most major CAD companies now developing tools to help mitigate variation in power consumption and circuit delay during the design phase.[1-3] Although many approaches claim excellent improvements in predicted chip yield, SSTA cannot mitigate lifetime degradation effects on delay and power without the up-front guard-banding cost. SSTA also suffers from scalability issues. Moreover, the body of work validating the benefits of SSTA is limited because thus far researchers have used only simulation and theoretical analysis, not silicon validation, to prove results.

Several circuit techniques for variability mitigation have emerged, including adaptive body bias, which has generated excellent results,[4] but with questionable scalability in advanced processes and in silicon-on-insulator (SoI) and multigate devices. Other techniques include adaptive supply voltage,[5] clock tuning,[6] and variation-tolerant keeper structures in rarely used dynamic circuits.[7] Using an array of these well-studied circuit techniques to tune individual processing elements (PEs) in a multicore system holds great promise for maximizing circuit performance and energy efficiency. You can apply these methods after initial fabrication and throughout the part's lifetime to address the effect of various degradation methods, such as electromigration, oxide wear-out, and negative bias temperature instability (NBTI).

Dynamic reliability management (DRM), which Srinivasan et al. first described,[8] and dynamic thermal management (DTM) techniques address performance degradation in ICs by trying to maintain predictable wear-out or temperature profiles up to a certain threshold. These techniques accomplish this goal using various actuators, such as dynamic supply voltage scaling or clock throttling.[9] Srinivasan et al. presented a multimechanism DRM method for single processors based on a sum-of-failures-in-time approach.[8] Their results focus on performance penalties of conservative thermal corners during design. Lu et al. presented a simpler DTM approach that aims to limit the on-die junction and wire temperature to guarantee predicted electromigration lifetime.[9] These earlier works considered single-processor systems.[10] However, highly parallel systems with many PEs can more effectively use the monitoring information from DRM or DTM to schedule heterogeneous elements in a multithreaded workload rather than limiting overall system performance.

References

1. C. Visweswariah et al., "First-Order Incremental Block-Based Statistical Timing Analysis," *Proc. Design Automation Conf.* (DAC 04), ACM Press, 2004, pp. 331-336.
2. H. Chang and S.S. Sapatnekar, "Statistical Timing Analysis Considering Spatial Correlations Using a Single PERT-Like Traversal," *Proc. Int'l Conf. Computer-Aided Design* (ICCAD 03), IEEE CS Press, 2003, pp. 621-625.
3. A. Agarwal, V. Zolotov, and D.T. Blaauw, "Statistical Timing Analysis Using Bounds and Selective Enumeration," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 9, Sept. 2003, pp. 1243-1260.
4. J. Tschanz et al., "Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, Nov. 2002, pp. 1396-1402.
5. J. Tschanz et al., "Effectiveness of Adaptive Supply Voltage and Body Bias for Reducing Impact of Parameter Variations in Low Power and High Performance Microprocessors," *IEEE J. Solid-State Circuits*, vol. 38, no. 5, May 2003, pp. 826-829.
6. P. Mahoney et al., "Clock Distribution on a Multi-core Multithreaded Itanium-Family Processor," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 05), IEEE Press, 2005, pp. 292-293.
7. C. Kim et al., "A Process Variation Compensating Technique for Sub-90nm Dynamic Circuits," *Proc. IEEE Symp. VLSI Circuits*, IEEE Press, pp. 205-206.
8. J. Srinivasan et al., "The Case for Lifetime Reliability-Aware Microprocessors," *Proc. 31st Ann. Int'l Symp. Computer Architecture* (ISCA 04), IEEE CS Press, 2004, pp. 276-287.
9. Z. Lu et al., "Interconnect Lifetime Prediction under Dynamic Stress for Reliability-Aware Design," *Proc. Int'l Conf. Computer-Aided Design* (ICCAD 04), IEEE CS Press, 2004, pp. 327-334.
10. E. Karl et al., "Reliability Modeling and Management in Dynamic Microprocessor-Based Systems," *Proc. 43rd Design Automation Conf.* (DAC 06), ACM Press, 2006, pp. 1057-1060.

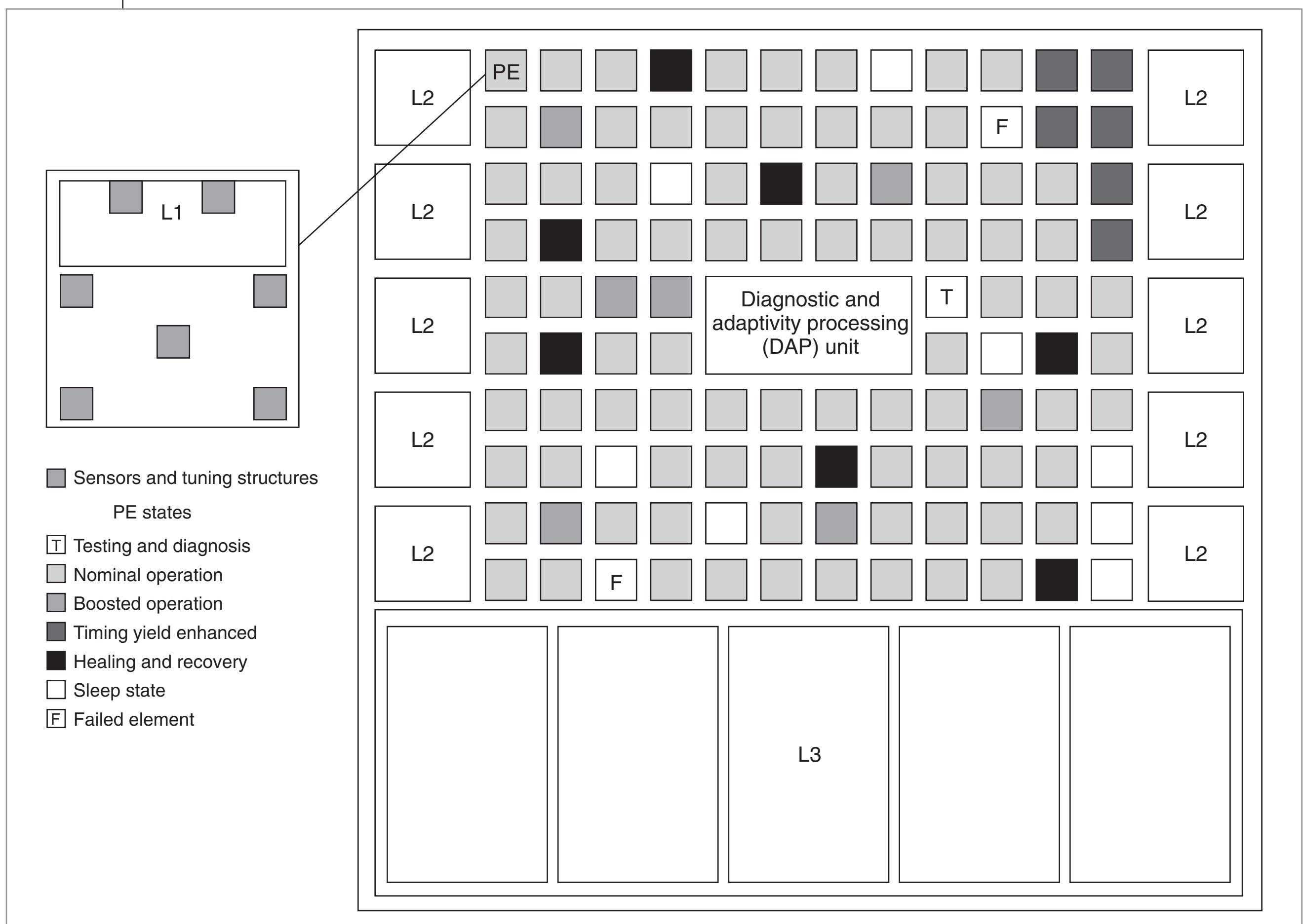

Figure 1. Overview of proposed ElastIC architecture, showing various processing-element operation modes. Each PE consists of a simple processor architecture with dynamic voltage scaling and tunable flip-flops.

mechanisms. It then initiates active healing, taking advantage of the reversibility of several wear-out mechanisms such as negative bias temperature instability (NBTI) and electromigration,[6,7] and tunes the PE for optimal compensation of parametric shifts. (The DAP unit's performance requirements are modest, so it can operate at low voltage and frequency, and it uses aggressive built-in redundancy, making it effectively immune to failure.)

Third, corresponding memory and interconnect systems use integrated error codes and redundancy to address functional failures and parametric shifts. Fourth, a system-level scheduler employs up-to-date reliability degradation and frequency or power trade-offs of individual PEs and memory elements to maximize global system performance under workload constraints.

Offline examination and rejuvenation

A key feature of ElastIC is that the DAP unit periodically takes PEs offline for detailed examination and rejuvenation. Many PEs share this unit, thus amortizing its overhead and making significant capabilities in the unit possible. The DAP unit provides the system-level scheduler with up-to-date visibility of each PE's power, performance, and expected lifetime trade-offs. Thus, a dynamic, tunable architecture like ElastIC lets you handle reliability and variation without using overly pessimistic margins for circuit timing and other performance metrics. Rather, the system-level scheduler manages reliability and variation dynamically and adaptively under comprehensive performance, power, and lifetime constraints.

Memory and communication structures and PEs

Reliable computing using unpredictable silicon requires fundamentally different approaches for memory and communication structures, as well as for PEs. Memory and interconnect restrict their operation to simple data storage and transmission and hence are

amenable to error detection through error codes. Furthermore, because of their regular structure, they address adaptivity using swappable spares. Both of these techniques are well established in memory systems with today's low manufacturing failure rates. Implementing this architecture requires reexamination of these techniques to address failures and parameter shifts at much higher levels. For example, the DAP unit will periodically examine error correction rates for memory and interconnect, and will issue a decree to replace frequently failing portions with spares. Furthermore, to address wear-out mechanisms that slow data transmission in buses, we can either enable spare drivers or use dynamic gate sizing through tunably sized drivers. Because monitored error rates rise with use, either of these alternatives is more desirable than employing a higher V_{DD}. In addition, we must address the interface of PEs, each running at its individual frequency and voltage, with ensuing contention and coherency issues.

Dynamic diagnosis and adaptivity

PEs' irregular structure makes using error codes and swappable spares ineffective and expensive. Hence, our first main principle for the ElastIC architecture is that it obtains robust operation of PEs through their inherent redundancy and through dynamic diagnosis and adaptivity. Our second principle is that exchanging one large, deeply pipelined processor for many small PEs, each with shallow pipelines, improves overall system reliability and predictability. A single failure disables only one of many PEs, and process variations average out over logic depth, so shallow pipelines with deep logic tend to exhibit less susceptibility to variation and parametric shifts. The optimal trade-off between pipelining, parallelism, and power-performance requires selecting the number of PEs and pipeline depth while maximizing reliability and predictability. The ElastIC architecture's third and central principle is that it addresses the impact of unpredictable silicon through the DAP unit's dynamic monitoring and tuning along with system-level scheduling.

Reliability monitoring, self-healing, and performance tuning

After taking a PE offline, the DAP unit employs in situ reliability, performance, and power sensors, along with predetermined ATPG vectors, to obtain individual observability of individual gates' or small gate clusters' reliability degradation and performance. For example, we can quantify the extent of oxide breakdown by monitoring oxide leakage current while simultaneously applying specific vectors to differentiate between gates in a circuit block. This enables mapping a PE's reliability characteristics and parametric shifts (or outright failure) with fine granularity.

Tracking the characteristics of individual devices or small clusters lets the DAP unit take three actions: First, it initiates active healing of particularly damaged clusters, taking advantage of the reversibility of several reliability effects. Examples include reversing currents in supply networks and interconnect to increase electromagnetic median time to failure (MTF) in targeted high-stress areas (which requires designing the power grid accordingly, incurring some acceptable overhead) or forcing periods of inactivity for portions of the PE to allow NBTI recovery.

Second, the DAP unit addresses detected delay shifts by tuning the processor with adaptive cycle-stealing elements built into the flip-flops. In this approach, as Figure 2 shows, the DAP unit controls short tunable periods of transparency in each flip-flop to provide an extra timing margin through cycle stealing on critical paths. This transparency makes it possible to average out delay variation in successive stages; hence, this approach is superior to intentional clock skew control—even adaptive clock skew control. Furthermore, we can cluster these flip-flops into sets, with each set sharing transparency controls, thereby reducing overhead to a feasible level. Adaptive transparency results in a direct trade-off between setup and hold time constraints, which the DAP unit can optimally control before placing the PE back online.

Third, the DAP unit conducts detailed performance and power characterization of a PE by testing functional operation at different clock frequencies and operating voltages. Using this data, the DAP unit generates a comprehensive model of reliability degradation, performance, and power trade-offs for use by the system-level scheduler.

System-level scheduling

Given the trade-off model, the system-level scheduler maximizes global system performance by optimally assigning each processor's voltage and frequency. In addition, in situ sensors monitor temperature for various architectural components—a necessary input for the underlying reliability and power models. The dynamic reliability scheduler makes it possible to limit voltage operation to the actual observed reliability degradation, given lifetime requirements. Dynamically setting the

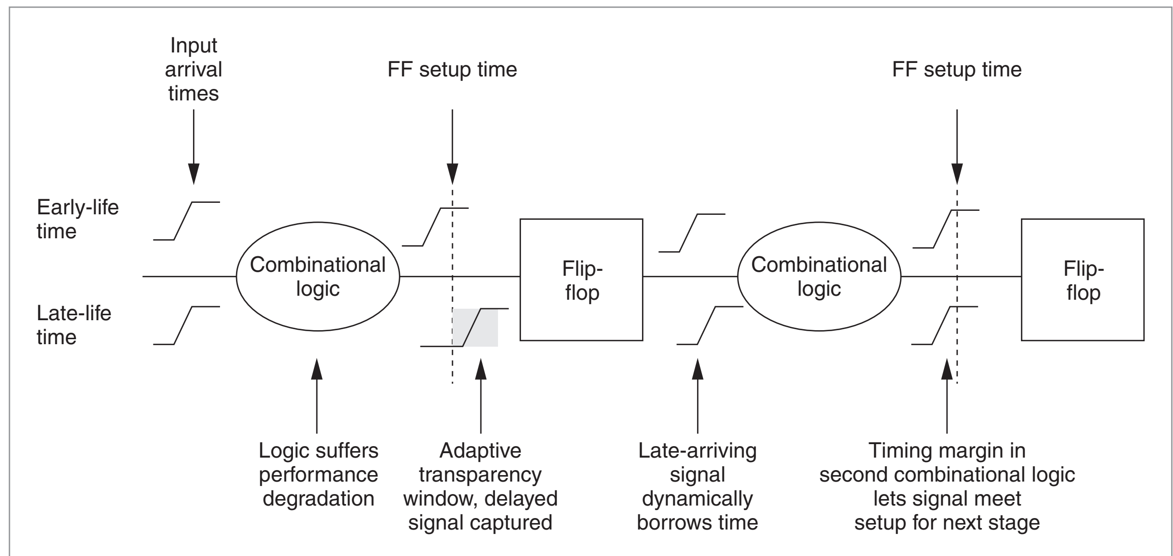

Figure 2. Benefit of tunable flip-flop transparency windows for maintaining performance under static and dynamic variability sources.

maximum limit on supply voltage operation exchanges conservative margins with dynamically managed reliability. The scheduler can also steer processor traffic on the basis of the behavior it learns about each PE—for instance, fast leaky cores and slow cores present different reliability, power, voltage scalability, and performance trade-offs. For example, tasks with low switching activities attain better energy efficiency when mapped to low-leakage PEs. On the other hand, we can scale the voltage for fast PEs more aggressively to yield acceptable leakage while maintaining reasonable performance.

Besides running PEs above their reliability capacity followed by periods of self-healing, ElastIC can leverage the novel concept of disposable transistors, which involves severely overstressing certain components for short periods of time to achieve otherwise impossible performance levels (thus, there are short periods of boosted performance in which device degradation averages over time to normal degradation). Using disposable PEs lets us schedule critical or sequential threads to PEs with a very short lifetime but ultrahigh performance. The goal is to have a system that schedules individual PEs disparately to maximize the capabilities of the highly unpredictable underlying silicon fabric, and to purposely drive the system to the limit at its end-of-life point.

Design and implementation

Other features of ElastIC that require careful investigation include

- employing several different architectures for the PEs rather than a single architecture,[8] which would broaden the power-performance design space and let the system-level scheduler more intelligently steer tasks to appropriate cores;
- using compiler directives to determine when to apply overdrive voltages, which can boost performance for portions of code having significant data dependencies or sequential bottlenecks that delay execution of subsequent parallelizable code fragments; and
- addressing soft errors with our proposed techniques as well as other techniques, such as extensions to Razor-style methods[9] and other efforts in this area—for example, work by Zhang and Shanbhag[10] and by Nicolaidis.[11]

Designing a highly adaptive architecture like ElastIC is challenging because there are no approaches that incorporate adaptive techniques directly into the design process. A new design optimization framework is necessary. So, we suggest an approach based on variation space sampling, which is ideally suited to adaptive microarchitecture-level techniques. This approach statistically samples the process space and then uses well-established, fast deterministic approaches to optimize the design at each sampled process condition. We can then construct probability distributions of the optimal design parameters to guide the optimization. Based on the successful use of this technique in other areas,[12] it

should also be valuable for architectural optimizations and clustering of the adaptive transparency flip-flops.

An extremely adaptive architecture obviously relies heavily on information gathered from the in situ monitoring circuits. Multiuse sensors that can characterize performance, degradation, and power consumption at fine granularities are critical to the ElastIC architecture. Striking the proper balance between estimation accuracy and area or power overhead is essential to providing value through this system-level approach to variability mitigation. Figure 3 illustrates this trade-off.

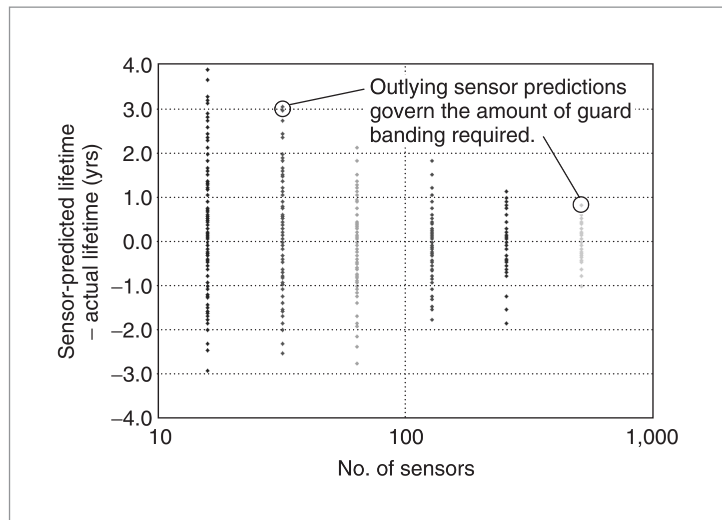

Figure 3. Diminishing difference between sensor-predicted lifetime and actual lifetime as sensor density increases. Using a four-level hierarchical variation model for each device's oxide thickness, threshold voltage, and channel length and width, this plot shows the effect of variation on sensors used for reliability management decisions based on oxide degradation. Increasing sensor density improves lifetime prediction and requires less guard banding for on-die voltage and temperature limits.

THE EXTREME ADAPTIVITY CONCEPT proposed here is applicable to other more mainstream (that is, not massively multicore) architectures. It's also possible to leverage many of the concepts such as active self-healing, soft-edge flip-flop tuning for enhanced timing yield, and dynamic reliability management (including disposable transistors) independently of the ElastIC architecture. Developing small, low-power sensors to be embedded throughout systems implementing the ElastIC architecture is a current focus of ongoing research. Analyzing the impact of process variations on sensor-based dynamic reliability management (DRM) is another area that merits further exploration. Understanding the behavior and mechanics of breakdown at the device and circuit levels for various wear-out mechanisms will allow meaningful new strategies to be developed at the functional block or system level, and aid the prediction-based models used in most DRM schemes. ■

■ References

1. J.M. Rabaey, "Design at the End of the Silicon Roadmap," *Proc. 10th Asia and South Pacific Design Automation Conf.* (ASP-DAC 05), ACM Press, 2005, pp. 1-2.
2. S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," *IEEE Micro*, vol. 25, no. 6, Nov./Dec. 2005, pp. 10-16.
3. T.C. Chen, "Where CMOS Is Going: Trendy Hype vs. Real Technology," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 06), IEEE Press, 2006, pp. 1-18.
4. V.D. Agrawal et al., "Built-in Self-Test for Digital Integrated Circuits," *AT&T Technical J.*, vol. 73, no. 2, Mar.-Apr. 1994, pp. 30-39.
5. Y. Zorian, E.J. Marinissen, and S. Dey, "Testing Embedded-Core-Based System Chips," *Computer*, vol. 32, no. 6, June 1999, pp. 52-60.
6. S. Rangan, N. Mielke, and E.C.C. Yeh, "Universal Recovery Behavior of Negative Bias Temperature Instability," *Proc. IEEE Int'l Electron Devices Meeting* (IEDM 03), IEEE Press, 2003, pp. 14.3.1-14.3.4.
7. C.-U. Kim et al., "Effect of Current Reversal on the Failure Mechanism of Al-Cu-Si Narrow Interconnects," *J. Electron Materials*, vol. 25, no. 2, Feb. 1996, pp. 293-296.
8. R. Kumar et al., "Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction," *Proc. 36th Ann. ACM/IEEE Int'l Symp. Microarchitecture* (Micro 36), IEEE CS Press, 2003, pp. 81-92.
9. D. Ernst et al., "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," *Proc. 36th Ann. ACM/IEEE Int'l Symp. Microarchitecture* (Micro 36), IEEE CS Press, 2003, pp. 7-18.
10. M. Zhang and N. Shanbhag, "A CMOS Design Style for Logic Circuit Hardening," *Proc. IEEE Int'l Reliability Physics Symp.*, IEEE Press, 2005, pp. 223-229.
11. M. Nicolaidis, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies," *Proc. IEEE VLSI Test Symp.* (VTS 99), IEEE CS Press, 1999, pp. 86-94.
12. S.H. Kulkarni, D. Sylvester, and D. Blaauw, "A Statistical Approach to Body Bias Clustering for Postsilicon Tuning," to be published in *Proc. Int'l Conf. Computer-Aided Design* (ICCAD 06), IEEE CS Press, 2006.

Dennis Sylvester is an associate professor of electrical engineering and computer science at the University of Michigan, Ann Arbor. He is also a visiting professor at the National University of Singapore. His research interests include low-power circuit design and design automation techniques, design for manufacturability, and on-chip interconnect modeling. Sylvester has a BS from the University of Michigan, Ann Arbor, and an MS and a PhD from the University of California, Berkeley, all in electrical engineering.

David Blaauw is an associate professor at the University of Michigan, Ann Arbor. His research interests include VLSI design and CAD, with emphasis on circuit design and optimization for high-performance and low-power applications. Blaauw has a BS in physics and computer science from Duke University, and an MS and a PhD in computer science from the University of Illinois, Urbana-Champaign.

Eric Karl is pursuing a PhD in electrical engineering at the University of Michigan, Ann Arbor. His research interests include low-power circuit design, and circuit and system design for reliability, variability, and manufacturability. Karl has a BS and an MS in electrical engineering, both from the University of Michigan, Ann Arbor. Karl is a student member of the IEEE.

■ Direct questions and comments about this article to Dennis Sylvester, 2417C EECS, 1301 Beal Ave., Univ. of Michigan, Ann Arbor, MI 48109-2122; dennis@eecs.umich.edu.

For further information on this or any other computing topic, visit our Digital Library at http://www.computer.org/publications/dlib.

Here Now!
Introduction to Python for Artificial Intelligence

By Steven L. Tanimoto
University of Washington

Python, an increasingly popular general-purpose programming language, offers a variety of features that make it especially well-suited for artificial intelligence applications. This ReadyNote will help professional programmers pick up new skills in AI prototyping and will introduce students to Python's AI capabilities. $19

www.computer.org/ReadyNotes