Reliable Systems on Unreliable Fabrics

Todd Austin, Valeria Bertacco, and Scott Mahlke
University of Michigan, Ann Arbor

Yu Cao
Arizona State University, Tempe

Editor's note:
Design integrity is becoming more difficult to achieve in advanced CMOS nodes. This article describes work in the Gigascale Systems Research Center that addresses this challenge, including both shorter-term stress-reduction techniques and longer-term design resiliency and verification efforts. Research in these areas is helping the industry move forward despite the design challenges at the nanoscale level.
—David C. Yeh, Semiconductor Research Corp.

The continued scaling of silicon fabrication technology has led to significant reliability concerns that are quickly becoming a dominant design challenge. Complexity challenges in the form of immense designs that defy complete verification, as well as physical challenges such as silicon aging and soft errors that impair correct system operation, are threatening design integrity.

The Resilient-System Design Team of the Gigascale System Research Center (GSRC) is addressing these challenges. Its synergistic research projects include developing near-term reliability stress reduction techniques to improve today's silicon quality, and exploring longer-term technologies to detect, recover, and repair faulty systems. These efforts are supported by research on fault modeling and functional-verification methodologies. The team's goal is to provide highly effective, low-cost solutions that ensure correctness and reliability in future designs and technology nodes, thus extending the lifetime of silicon technologies beyond what currently appears profitable. This article describes four of the team's projects.

The bumpy road ahead for silicon

As silicon technologies move into the nanometer regime, designers must contend with a barrage of reliability threats that lead to potentially unreliable components and design processes. These threats arise from various sources:

- *Silicon failures and noise effects.* Shrinking lithography and lower design tolerances make ICs increasingly susceptible to manufacturing defects, leading to substantial problems in maintaining economically viable chip yields. With rising power density and nonideal threshold and supply voltage scaling, wearout-related permanent errors and transient soft errors become increasingly common during a chip's lifetime. Transient errors are bound to occur at perceivable rates in the growing market of digital-analog-RF integrated circuitry. In emerging data transmission protocols involving multi-GHz wireless communications systems, channel noise and interference will limit the rate of reliable data transmission. For memory systems, random noise effects, such as those induced by radioactive particles or telegraph noise, increasingly jeopardize data stability, especially in low-voltage circuits.
- *Extreme process variation.* Subwavelength lithography leads to large variation in transistor geometries and flat-band voltage. Nanoscale devices are prone to large spatial variation in threshold voltage caused by intrinsic aspects such as line-edge roughness, random dopant fluctuations, and body thickness variations in silicon-on-insulator devices. Such variation can lead to large spreads in circuit delay, power, and robustness for digital ICs, and in performance for analog ICs.
- *Design errors.* The growing complexity of system hardware is defeating the enormous effort put forth by verification engineers to ensure system correctness. The reason is that verification is unable to keep up with modern high-productivity

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

design solutions such as SoCs, which feature multiple complex hardware and software components connected by a diverse set of interfaces. Today, most, if not all, complex system designs are released containing latent bugs, which sometimes become evident after a design reaches the market.

Ultimately, these challenges threaten the continued scaling of silicon fabrication technologies. A primary goal of transistor scaling is to reduce the cost of electronic devices. As devices scale to smaller geometries, however, they become less reliable, necessitating the inclusion of reliability mechanisms. Reliability costs range from service and replacement to built-in solutions entailing area and design resources. These costs are increasing at technology nodes with higher natural failure rates, which require more robust and finer-grained reliability techniques. As Figure 1a shows, the financial impact of reliability infrastructures will eventually make CMOS-based silicon scaling economically unfeasible. Figure 1a shows current cost trends of transistor fabrication and built-in or service-based reliability solutions. Figure 1b shows how low-cost built-in reliability techniques can lower the overall cost of silicon products.

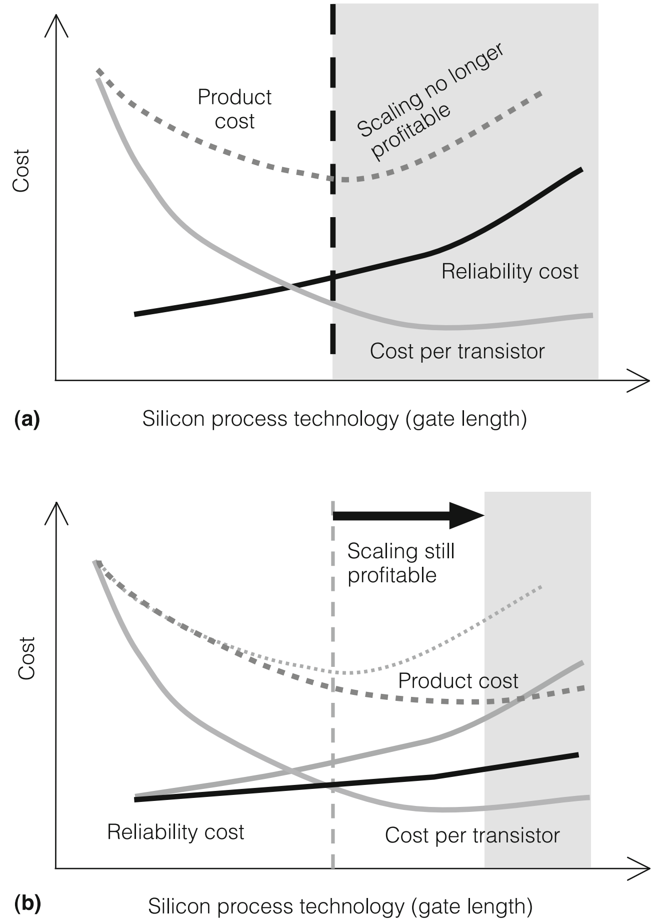

Figure 1. Silicon process technology trends. The inherent variability and high failure rates of transistors in near-future technology nodes will demand high-cost reliability service and solutions, bringing the production of digital electronic devices in traditional silicon designs to unacceptable costs (a). The development of new low-cost, resilient designs will make further transistor scaling economically viable, extending the lifespan of CMOS-based silicon (b). In (a), reliability cost = failure rate × (replacement costs and area and design costs of defect-tolerant mechanisms). In (b) reliability cost = failure rate × low-cost built-in defect-tolerant mechanisms.

Toward reliable silicon fabrics

One key GSRC research area is resilient-system design. The research group at work on this topic is investigating the problem broadly, from quantifying functional and physical reliability threats to developing customizable, extensible, and cost-effective design methodologies. At the core of the effort is the investigation of solutions incorporating vertically integrated technologies that draw from circuit, microarchitecture, and software innovations and are applicable to both processors and heterogeneous front ends. We strive to develop flexible solutions that are easily adaptable to different applications and system domains, and that are extendible to other reliability issues such as software defects.

Deploying resiliency mechanisms is often a challenge, especially if they are being deployed before the onset of the failures they target. Too often, designers perceive the reliability infrastructure as a burdensome tax to be paid only as a last resort in accomplishing dependability goals. However, if designers can leverage a resiliency mechanism to further enhance system value, the additional benefits it brings can offset its cost and burden. Recent GSRC reliability projects, including the Razor and Algorithmic Noise Tolerance projects,[1,2] leverage resiliency to lessen power demands. Likewise, the Reliability and Security Engine project is developing a processor-level framework that provides security guarantees along with transparent application-specific reliability.[3]

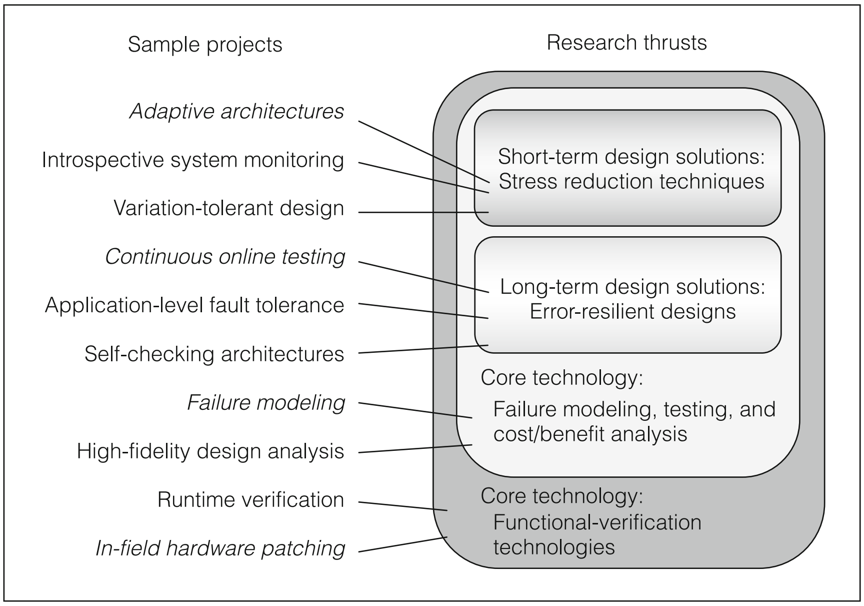

Figure 2. Structure of the GSRC Resilient-System Design Team's research thrusts. The team is organized into four research thrusts: The two core technology areas develop component solutions and technologies that are used by the other two thrusts to provide near- and long-term system solutions.

The GSRC Resilient-System Design Team includes 16 primary investigators, who are faculty members at 10 universities. As Figure 2 shows, the team is organized into four main research thrusts. Two thrusts focus on cross-cutting solutions and infrastructure: the exploration of high-level modeling, testing, and analysis technologies; and runtime and in-field functional verification methodologies. The other two thrusts build on the first two to provide complete system solutions for short-term stress reduction techniques and longer-term error-resilient mechanisms. The former provide solutions to issues already present in today's silicon designs; the latter are dedicated to mid- and long-term issues of future silicon generations. Figure 2 also lists sample projects under investigation by the team. Here, we briefly describe the research in each thrust.

Stress reduction techniques

Stress factors, including high temperatures, voltages, and switching activity, exacerbate the system reliability challenge. Thus, one of the team's goals is to develop low-cost monitoring techniques to dynamically quantify the stress that a system is experiencing. Quality-of-service scheduling techniques make it possible to use system-level control to reduce overall system stress, slowing down the inevitable march toward system failure. A few initial solutions focus on active temperature management for chip multiprocessors (CMPs) through dynamic resource assignment. In addition, similar system-level control techniques can sometimes reverse wearout factors such as negative-bias temperature instability (NBTI).

Error-resilient designs

For some environments, system survivability and safety require facing faults head on. Error sources such as gate wearout, design errors, and metal electromigration can cause failures that must be detected and corrected to keep the system operational. For a system to survive hard-failure events, it must include diagnostic capabilities to locate the errant component, and a repair mechanism to rehabilitate the system to continue computation. The Resilient-System Design Team focuses on developing error detection, correction, recovery, and repair technologies for hardware and software components of digital, mixed-signal, and mixed-technology systems. The team targets the development of system-level checkers, online testing and verification techniques, fault diagnosis, and repair and recovery capabilities.

Failure modeling and cost-benefit analysis

To enable the development of research solutions in the two previous thrusts, we are developing failure models that assess the likelihood of a fault's manifesting itself and the cost of protecting against it. Accurate failure modeling helps us understand which failures we need to address and which fault tolerance techniques provide the best protection. Cost-benefit analysis helps designers determine whether including a specific resilient-design technique will lead to a cheaper component. We are also developing a scaling theory that incorporates reliability along with traditional metrics such as area, performance, and power.

Functional verification technologies

The second core thrust investigates error detection mechanisms through design-time (for functional errors) and runtime (for silicon failures and functional errors) verification solutions. As mentioned earlier, the growing complexity of digital designs has created a verification gap that allows all modern designs to be released with latent bugs. This thrust develops verification technologies that scale more effectively, including hybrid-analysis solutions (such as semiformal verification) and higher-level and semantically rich design languages that facilitate automated reasoning about design correctness. Additionally, this thrust researches runtime verification solutions that can correct design errors that have escaped into silicon directly in the field. These solutions include the addition of hardware checkers or hardware-patching mechanisms.

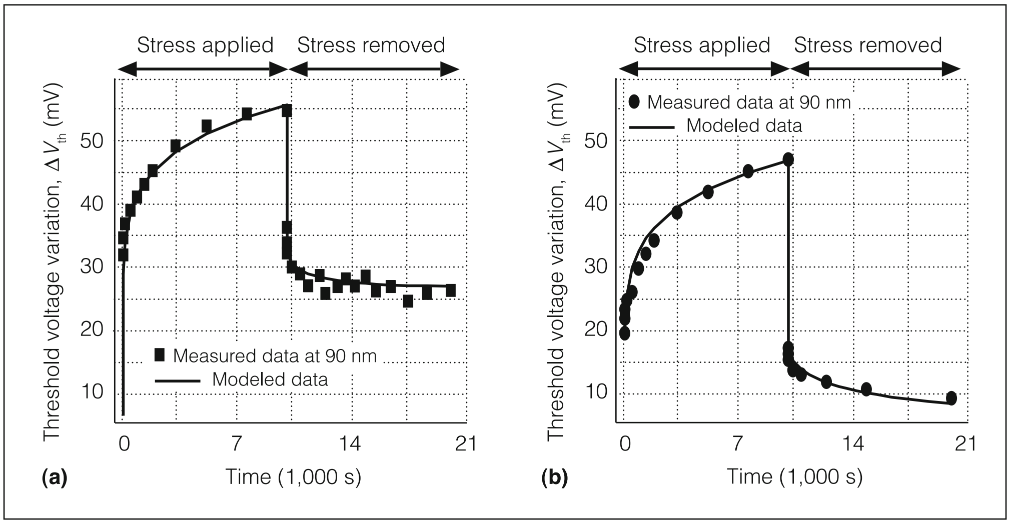

Figure 3. NBTI transistor-level effects: T_{ox} = 1.2 nm (a); T_{ox} = 2.2 nm (b). In the first half of the experiment, a voltage stress is applied to a transistor, causing an elevation of V_{TH} due to NBTI. After removal of the stressing voltage, the recovery effect heals NBTI damage over time. Our models capture these effects and closely follow measured data.

Research in action

Here, we describe four projects developed in the Resilient-System Design research effort, one from each research thrust.

Fault modeling and resiliency analysis

Aggressive scaling of CMOS technology inevitably leads to new reliability problems, such as the NBTI prominent in PMOS devices and time-dependent dielectric breakdown. These effects usually manifest as a temporal degradation of circuit performance (that is, aging). To cope with these threats, traditional research has focused only on improving process technology, and VLSI designers rely on simple guard banding to bypass the analysis and optimization of these time-dependent effects. However, with reliability degradation exacerbated by continual scaling, a conservative design approach will cause excessive overmargining. To improve circuit reliability and design predictability, it is critical to develop reliability models that address emerging failure modes, and new tools that predict and diagnose circuit performance degradation.

To achieve these goals, we recently developed an integrated modeling framework that accurately characterizes two leading reliability problems: NBTI and channel hot-carrier (CHC) effects.[4] Both effects induce interface charges in gate oxide, increase the threshold voltage, and reduce mobility. Consequently, they degrade transistor drive current, circuit speed, noise margin, and matching property. Whereas CHC occurs only during dynamic switching, NBTI occurs under static stress and is the limiting factor of circuit life in modern microprocessor and SoC designs.

Our unified reliability model is based on the general reaction-diffusion mechanism.[4] The model captures the two steps that lead to circuit degradation. The first step is reaction: Si-H or Si-O bonds at the interface of the Si substrate and gate oxide break under electrical stress. This results in the generation of interface charges. The second step is a process of diffusion of species generated from the reaction away from the interface. As a result, the degradation rate of the threshold voltage V_{TH} has a power-law dependence on the stress time: $\Delta V_{TH} \propto Kt^n$, where n is about 0.1 to 0.3.[4] K is an exponential function of temperature, voltage, and gate-oxide thickness due to the reaction process.

A unique property of NBTI is the recovery effect. As Figure 3 shows, during dynamic circuit operation, the degradation of V_{TH} can be partially recovered and circuit lifetime restored. Using these observations, the team developed a transistor-level reliability model to accurately predict such a recovery effect.[5] Depending on the duty cycle, appropriately biasing the PMOS gate can heal more than 75% of previous NBTI-induced degradation. Such behavior indicates that duty cycle control is very effective for minimizing circuit-aging effects of NBTI.

To further explore design techniques for circuit reliability, the Resilient-System Design Team will implement these degradation models in circuit simulation tools to diagnose their impact on dynamic and static circuit operation, identify the critical functional units, and evaluate trade-offs among design choices. Analyzing circuit resilience is particularly challenging because it involves time dependence that changes with node activity, input patterns, voltage, and temperature. Usually, these parameters are not spatially or temporally uniform but vary significantly from gate to gate and over time, owing to the uncertainty in operations. On the other hand, we have demonstrated that it is feasible to reliably predict the bound of circuit timing degradation under various input conditions.[6] Based on an aging-aware library, the proposed method recognizes a peak duty cycle (PDC) for each path, which leads to the maximum timing degradation. The exact PDC value depends on the logic topology and is correlated from path to path. By calibrating the upper bound of circuit aging, this approach avoids overly pessimistic guard bands in optimizing the design. Furthermore, it helps predict the relative importance of circuit units under increasingly severe circuit failure. Prediction of critical units is extremely important for future resilient design because it enables efficient testing and adaptive protection with minimal area and latency penalties. The modeling and prediction from these tools will provide early comprehension of failure mechanisms and design opportunities.

Introspective stress reduction via StageNet

Research into the physical effects of wearout on circuits has shown that many wearout mechanisms in silicon devices are progressive over time. These mechanisms, such as oxide breakdown, hot-carrier injection (HCI), and NBTI, negatively affect device performance. For example, a device subject to HCI experiences drive-current degradation, which leads to decreased switching frequency. The recognition of progressive performance degradation as a precursor to wearout-induced failures creates a unique opportunity for introspective reliability management techniques—that is, techniques that anticipate failures by leveraging dynamic timing analysis of logic in situ.

The StageNet research project focuses on tracking the progression of wearout through a CMP system's lifetime, and adapting both the architecture configuration and the system utilization to the dynamic circuit behavior.[7] StageNet has two main objectives: to create a CMP system that can tolerate 100 device failures while still delivering 90% of the promised system performance over its lifetime, and to postpone (or avoid altogether) the emergence of such failures through intelligent reliability management. Rather than performing online diagnostics and reacting to failures, StageNet takes a proactive approach to reliability management. Proactive management enables StageNet to adapt to process variation or excess stress on a system subset, thus maximizing performance over the entire operational lifetime.

As Figure 4 shows, three tightly integrated technologies make up the StageNet system: in situ wearout sensors, an introspective reliability management software layer, and an adaptive architecture fabric.

In-situ sensors. StageNet uses an online latency-sampling unit, the wearout detection unit (WDU), to measure propagation latencies for signals in the microprocessor logic. A statistical-analysis mechanism then samples and filters this information, accounting for anomalies in the sample stream (caused by phenomena such as clock jitter and power and temperature fluctuations). Thus, the WDU can identify significant changes in a given structure's latency profile and predict a device failure. Online statistical analysis allows the WDU to be self-calibrating, adapting to each structure that it monitors, so it is generic enough to be reused for various microarchitectural components.

Introspective reliability management. StageNet accomplishes reliability management using a runtime software layer located just below the operating system. This reliability manager receives chipwide sensor data and analyzes short- and long-term trends indicated by the data to create a reliability assessment of all components in the system. It uses TRIX, a trend analysis technique for measuring momentum in financial markets, to analyze circuit performance and to project future trends. TRIX relies on the composition of three calculations of the EMA (exponential moving average). The EMA is calculated by combining the current sample value with a fraction of the previous EMA, causing the weight

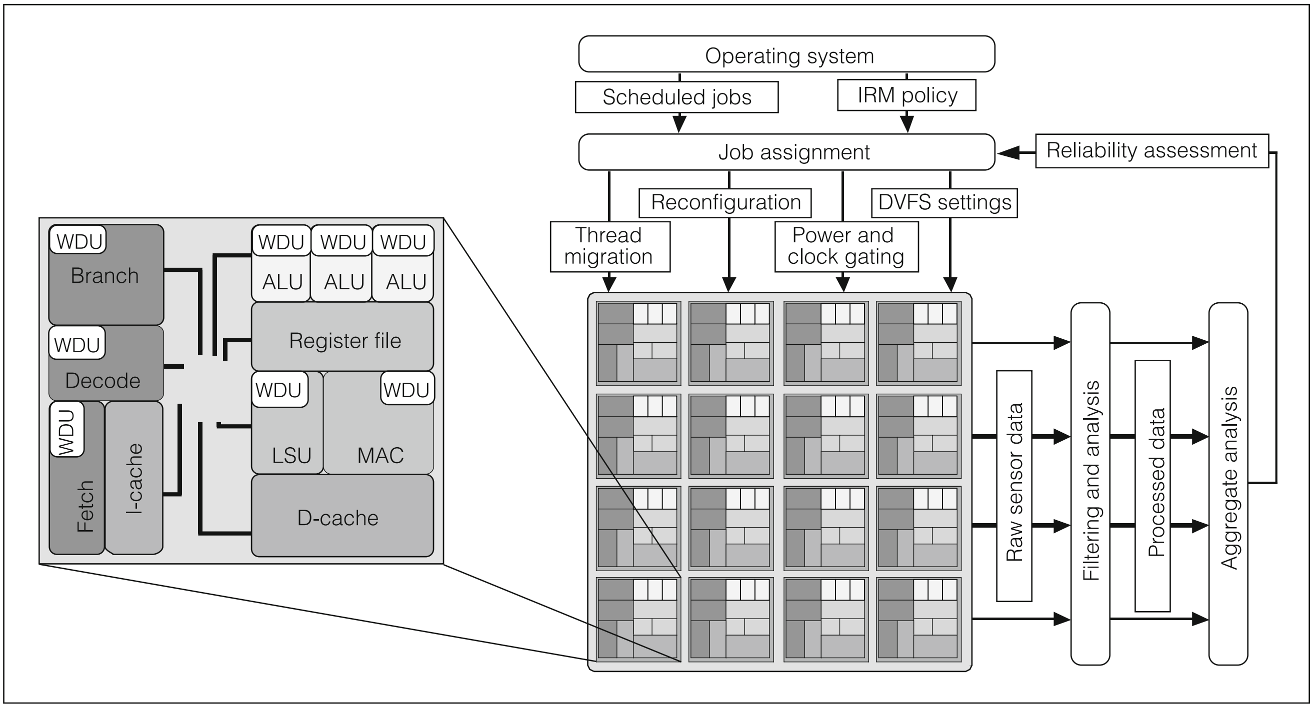

Figure 4. Introspective reliability management (IRM) in the StageNet multiprocessor system. The three central components are the wearout detection units (WDUs) associated with each processor module (shown on the left); the reliability manager, which analyzes sensor data to assess the reliability of the system and to control the assignment of jobs onto the processors (shown on the top and right); and the underlying architectural fabric, consisting of a network of coarse-grained processor pipeline stages (shown on the bottom). (ALU: arithmetic logic unit; DVFS: dynamic voltage and frequency scaling; LSU: load-store unit; MAC: multiply-accumulate.)

of older sample values to decay exponentially over time. The job assignment module uses the trend analysis to guide thread assignment and migration as well as dynamic voltage and frequency scaling. By incorporating direct measures of the circuit performance provided by the sensors, StageNet monitors damage from process variation and age-related degradation in real time.

Adaptive architecture fabric. Graceful performance degradation is essential to maintaining operation in the presence of wearout faults. StageNet uses a more fine-grained approach for reconfigurability and redundancy to maximize the computing resources available over the lifetime of a system. StageNet is a multiprocessor architecture that uses the pipeline stages of a traditional microarchitecture as the unit of replication. Stages, organized as a tightly coupled, high-performance network on a chip, communicate with one another through an interconnection rather than pipeline latches, allowing a high degree of system reconfigurability. The objective of this design is a scalable, fault-tolerant multiprocessor system with built-in redundancy and reconfiguration capabilities. Figure 4 shows a simple configuration of two coarse-grained stages, but a design can also be broken into finer-grained stages.

The key insight behind StageNet is that as silicon technologies continue to scale, resiliency must be a primary computer system design constraint. These systems must dynamically adapt their behavior to account for high levels of process variation and device wearout early in the product lifetime. Although system performance will degrade over time, resource availability must be maximized to ensure that the promised performance level is maintained.

Error-tolerant designs

As the severity of silicon failure increases, it will become necessary to incorporate mechanisms into the design that can detect occurring failures, recover system state, and repair underlying hardware components to allow continued system operation. With the growing challenges of producing reliable components

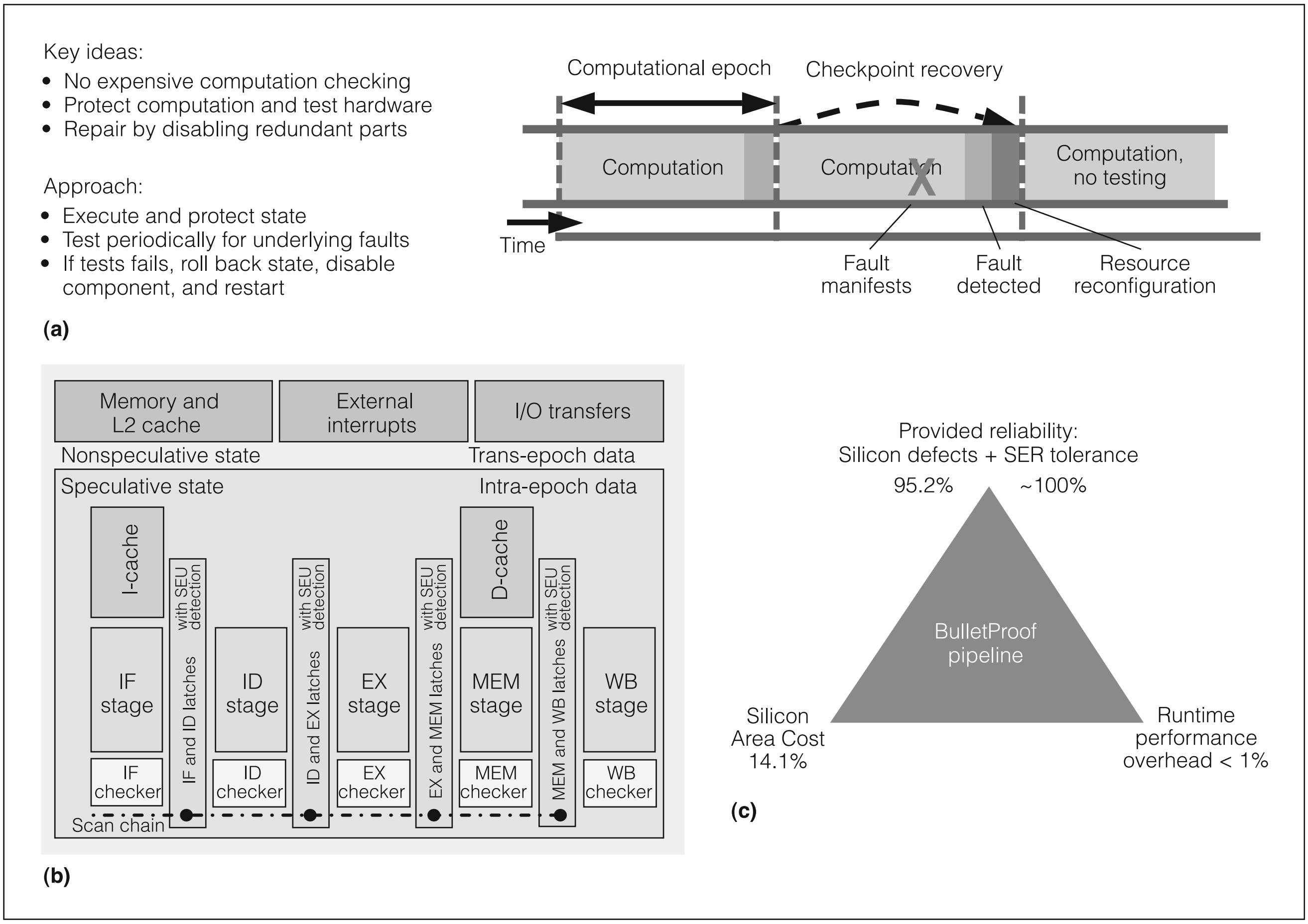

Figure 5. BulletProof defect-tolerant pipeline: overview (a), system architecture (b), and measured results (c). Periodic online testing identifies defects at runtime. Microarchitectural checkpointing mechanisms implement system recovery. The technique provides high defect coverage with minimal impact to system cost and performance. (EX: execute; ID: instruction decode; IF: instruction fetch; MEM: memory access; SER: soft error; SEU: single-event upset; WB: writeback.)

in extremely dense silicon technologies, this time is probably not far off.[8] Our BulletProof Project focuses on developing system-level checkers; online testing and verification techniques; and fault diagnosis, repair, and recovery capabilities for digital systems' hardware and software.[9]

The BulletProof pipeline is low-cost technology that protects a processor pipeline and its cache memory system from transient faults and permanent silicon defects. As Figure 5 shows, BulletProof keeps costs low through a combination of online distributed checkers and microarchitectural checkpointing, which efficiently identify defects and enable recovery from their impact. The online checkers periodically inject test vectors into the underlying hardware to verify that it can function correctly. If the online tests succeed, then the underlying hardware is defect free, and the previous checkpoint is no longer needed. If a checker detects a defect, a microarchitectural checkpointing mechanism restores processor state by rolling back execution by up to hundreds of thousands of cycles. Once the system is restored to the last checkpoint, the pipeline control repairs the hardware by reconfiguring it to operate without the defective component, possibly with a slight performance degradation. We can keep repair costs to essentially nothing by utilizing the natural redundancy of instruction-level parallel processors and multicore processors. For these designs, we simply disable the defective pipeline or processing element.

To protect the pipeline from intermittent transient faults and latch defects, we use a double-sampling latch design. The design presented in this article is based on a reliable pipeline solution we presented earlier.[1] It works by sampling all pipeline stage outputs twice—once at the prescribed clock period, and once again slightly into the next clock cycle. If a transient fault is exposed, it will produce a glitch at the latch input. Given the short duration of transient glitches,[10] only one of the samples will capture the errant result. Thus, if the samples differ, a transient fault has occurred. This event can be recorded in a single bit that can be probed at the end of checkpoints, in which case the system state can be rolled back and reexecuted.

The BulletProof approach provides high fault coverage with very low area and performance costs. It achieves low cost by avoiding traditional mechanisms (such as dual-modular redundancy) that replicate hardware (or computation) to validate results. These approaches start with overheads of 100% and quickly rise from there. In contrast, BulletProof detects the underlying failure mechanisms that impair correct computation (hardware defects or logic glitches) at a far lower cost than the resultant errant computation. In a recent report, we extensively evaluated the effectiveness, performance overhead, and area overhead of our defect tolerance technique extended to very low-cost software-based defect testing.[9] Implemented on a commercial CMP based on Sun's Niagara, the technique provided defect tolerance for 99.2% of the chip area with only a 5.8% area overhead. Through cycle-accurate simulation, we showed that this level of defect coverage incurred an average performance slowdown of only 5.5%.

Focus on correctness

Manufacturers and design houses alike strive to validate and verify their designs as much as possible during system development and then to contain the impact of escaped bugs through publicly available errata reports. Some escaped bugs are innocuous and easily overcome through a basic I/O system or operating-system update. But others are potentially harmful to system users, system security, or system performance.

We propose attacking this challenge by managing and partitioning the verification effort between presilicon verification and post-silicon runtime error detection and correction. We base this partitioning on the time and effort available for presilicon verification. In presilicon, it is critical to address all the system configurations (or design states) which occur frequently at runtime. This is the task at which presilicon validation is most effective. Frequently occurring configurations have many more opportunities to appear and be analyzed in the relatively small number of clock cycles that a logic simulator can execute. In contrast, we address the verification of all other design states—the rare configurations—as soon as they occur at runtime through a hardware-patching mechanism called field-repairable control logic (FRCL).[11] This mechanism performs in-field correction of errors in a design's control logic.

Most processor manufacturers maintain a specification update document reporting all bugs that become known after product release. From a comparative analysis of such reports, we found that most escaped bugs are due to design errors in the processor's control logic and very often to interference between multiple instructions ''in flight'' at the same time. Hence, in developing the FRCL mechanism, we specifically targeted errors related to dependency and correlation between instructions. In contrast, we assume that the system's data path is operating correctly. In fact, their modularity makes data path components less prone to error, and solutions (microcode-patching techniques) are available in the industry to correct functional errors exposed by individual instructions.

FRCL detects and corrects functional design errors at a performance cost. As long as the error frequency is sufficiently small, the performance impact remains imperceptible (less than 5% performance impact for an error frequency below one per 1,000 instructions). In addition, the mechanism's area cost is extremely small (less than 0.1%). Its hardware components include a state matcher and a reconfiguration manager block. We route signals from the processor's critical-control state to the state matcher, and program it to identify erroneous configurations. When the matcher finds a flawed configuration, the processor switches to a degraded-operation mode, which excludes most system features and is simple enough to be formally verified, yet can still execute the full instruction set architecture, one instruction at a time. After the program segment exposing the design flaw has executed in degraded mode, we can switch the processor back to full-performance mode.

We designed FRCL to handle flaws in processor control circuitry for components already deployed in the field. When customers detect an escaped error, they send a report containing the error description, such as the sequence of executed operations and the

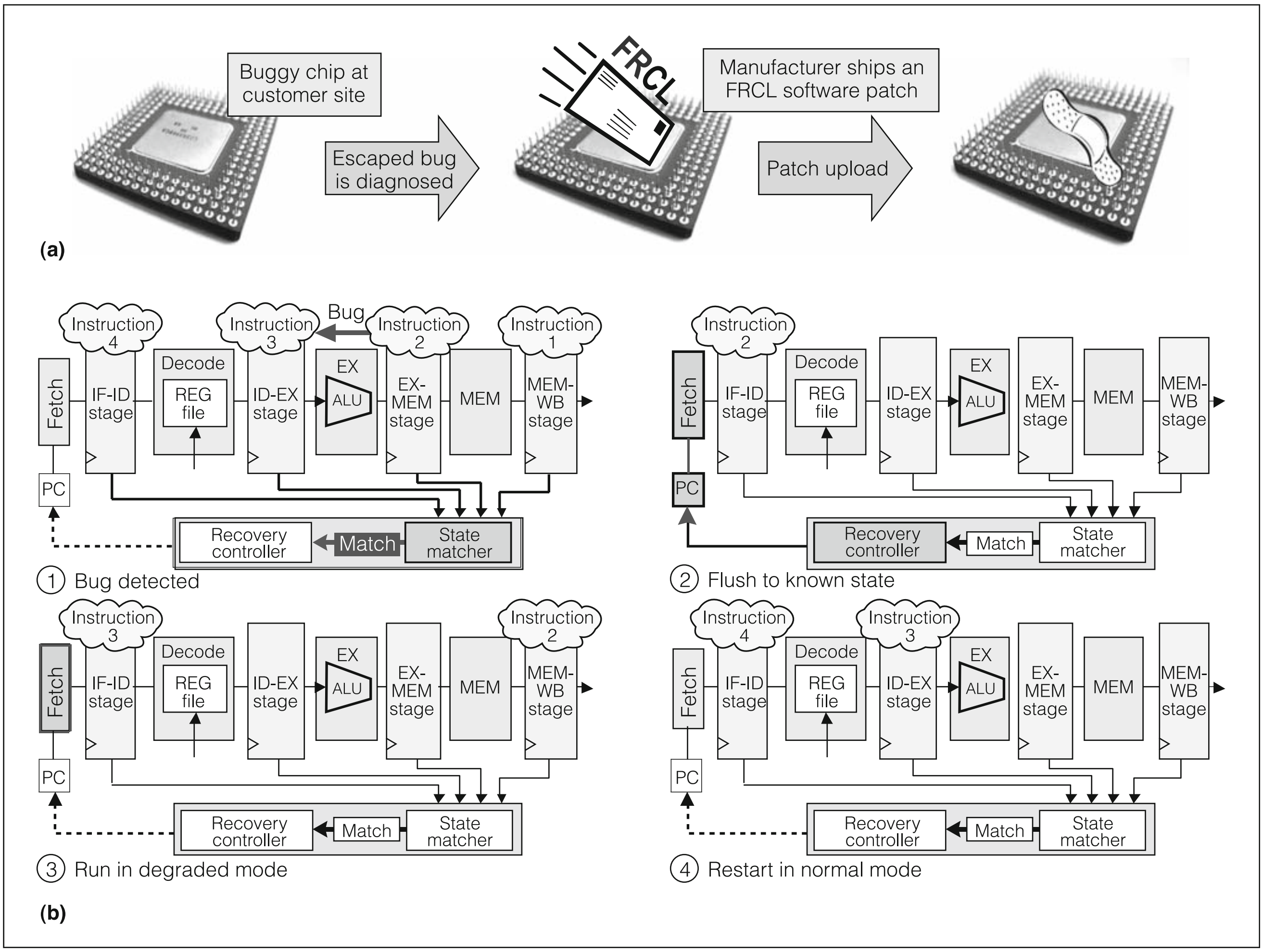

Figure 6. Field Repairable Control Logic is a solution to correct functional errors in processors already deployed in the field. FRCL enables a manufacturer to modify the system's functionality over its lifetime, thus making hardware as malleable as software. FRCL hardware patches are uploaded dynamically in the state matcher, updating and correcting the system's behavior (a). In FRCL's dynamic execution flow, a matcher continuously monitors the system's critical control state, and upon detection of a bug the pipeline is flushed, and execution advances in the fully verified degraded mode until the erroneous state is overcome (b). (PC: program counter; REG: register.)

values in the status registers, to the design house. Product support engineers investigate the problem, identify the error's root cause and the products affected by it, and decide on a way to correct it. Solutions include instruction or microcode patching. However, these can have a very high performance overhead or cost. In addition, they can only address errors that manifest on individual instructions, so they are difficult to deploy on instruction-interdependent errors.

Figure 6 shows our solution as implemented in FRCL. Based on knowledge of the bug's cause and which signals the matcher monitors in the defective processors, the engineering team can create patterns that describe the flawed control state configuration. They can then send the patterns to the customer as a patch, which is uploaded into the state matcher at startup. Every time the processor encounters the patched error at runtime, recovery begins, restoring the correct processor state and advancing program execution to effectively bypass the bug.

Our recovery mechanism requires no additional hardware or processor watchdog. Instead, we identify a subset of the processor's components that is functionally complete and, at the same time, simple enough to be formally verified. We call this subset the inner core because it is contained within the microprocessor core and is a core in itself. From another standpoint, the inner core is a defeatured version of the full processor. It mostly includes data path components; excluded features are most of the performance-enhancing units, such as pipelining, data forwarding, and branch prediction units. Only one instruction can execute at any given time in the inner core. This feature alone is easy to implement by reconfiguring the fetch unit, but it automatically excludes most of the complex control logic blocks that deal with instruction interdependency. During normal execution, when a buggy state is detected, the pipeline is flushed and the processor switches to its inner core. Because this mode is formally verified at design time, we can rely on it to correctly complete the next instruction.

Finally, the high-performance operation mode is restored. Completion of only one instruction is sufficient before normal operation resumes: In the event that the pipeline again steps into an error state, it will again enter the degraded mode to complete the following instruction. On the other hand, a designer can choose to run in degraded mode for several instructions to guarantee that the bug is bypassed entirely in a single recovery.

We are investigating several design trade-offs. For example, we are deciding which and how many signals the error-monitoring block should observe. We are also looking for solutions that can prevent a system from entering any potentially buggy state (a state that hasn't been verified at design time), thus providing correct execution even with an underlying buggy design.[12]

THE SUSTAINED PUSH toward smaller technology sizes has reached the point where device reliability has moved to the forefront of concerns for next-generation designs. If not addressed cost-effectively, silicon failure mechanisms threaten the yield and product lifetime of future systems. Current design paradigms assume that no gate or interconnect will ever fail during a system's lifetime. Additionally, existing fault tolerance techniques, such as classical redundancy and core sparing, are too expensive and unresilient to handle these problems. At the same time, the growing complexity of hardware and software verification has outstripped the efforts of verification engineers and academic researchers, putting the reliability of future designs at risk. The GSRC Resilient-System Design Team researchers are working together to overcome these challenges. Their effort to develop modeling techniques, resiliency mechanisms, and verification methodologies promises to extend the useful lifetime of scaling silicon technologies. ■

Acknowledgments

We acknowledge the support of the Gigascale Systems Research Center (GSRC). This work (and our related work) is also supported by grants from the National Science Foundation and Semiconductor Research Corp.

■ References

1. T. Austin et al., "Making Typical Silicon Matter with Razor," *Computer*, vol. 37, no. 3, Mar. 2004, pp. 57-65.
2. R. Hegde and N.R. Shanbhag, "Energy-Efficient Signal Processing via Algorithmic Noise-Tolerance," *Proc. Int'l Symp. Low-Power Electronics and Design* (ISLPED 99), ACM Press, 1999, pp. 30-35.
3. N. Nakka et al., "An Architectural Framework for Providing Reliability and Security Support," *Proc. Int'l Conf. Dependable Systems and Networks* (DSN 04), IEEE CS Press, 2004, pp. 585-594.
4. W. Wang et al., "An Integrated Modeling Paradigm of Circuit Reliability for 65 nm CMOS Technology," *Proc. Custom Integrated Circuits Conf.* (CICC 07), IEEE Press, 2007, pp. 511-514.
5. R. Vattikonda, W. Wang, and Y. Cao, "Modeling and Minimization of PMOS NBTI Effect for Robust Nanometer Design," *Proc. 43rd Design Automation Conf.* (DAC 06), ACM Press, 2006, pp. 1047-1052.
6. W. Wang et al., "An Efficient Method to Identify Critical Gates under Circuit Aging," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design* (ICCAD 07), 2007, pp. 735-740.
7. J. Blome et al., "Self-Calibrating Online Wearout Detection," *Proc. 40th Ann. IEEE/ACM Int'l Symp. Microarchitecture* (Micro 07), IEEE CS Press, 2007, pp. 109-120.
8. S. Borkar, T. Karnik, and V. De, "Design and Reliability Challenges in Nanometer Technologies," *Proc. 41st Design Automation Conf.* (DAC 04), ACM Press, 2004, p. 75.
9. K. Constantinides et al., "Software-Based Online Detection of Hardware Defects: Mechanisms, Architectural Support, and Evaluation," *Proc. 40th Ann. IEEE/ACM Int'l Symp. Microarchitecture* (Micro 07), IEEE CS Press, 2007, pp. 97-108.
10. S.S. Mukherjee, J. Emer, and S.K. Reinhardt, "The Soft Error Problem: An Architectural Perspective," *Proc. 11th*

Int'l Symp. High-Performance Computer Architecture (HPCA 05), IEEE CS Press, 2005, pp. 243-247.

11. I. Wagner, V. Bertacco, and T. Austin, "Shielding against Design Flaws with Field Repairable Control Logic," *Proc. 43rd Design Automation Conf.* (DAC 06), ACM Press, 2006, pp. 344-347.
12. I. Wagner and V. Bertacco, "Engineering Trust with Semantic Guardians," *Design, Automation and Test in Europe Conf.* (DATE 07), IEEE CS Press, 2007, pp. 140-145.

Todd Austin is an associate professor of electrical engineering and computer science at the University of Michigan, Ann Arbor. His research interests include reliable-system design, verification, and performance analysis tools. He has a BS in electrical engineering from the University of Wisconsin–Madison, an MS in computer engineering from Rochester Institute of Technology, and a PhD in computer science from the University of Wisconsin–Madison.

Valeria Bertacco is an assistant professor of electrical engineering and computer science at the University of Michigan, Ann Arbor. Her research interests include formal and semiformal design verification, with emphasis on full design validation and digital-system reliability. She has a Laurea in computer engineering from the University of Padua, Italy, and an MS and a PhD in electrical engineering from Stanford University. She is a member of the IEEE and the ACM.

Scott Mahlke is an associate professor in the Electrical Engineering and Computer Science Department at the University of Michigan, Ann Arbor, where he directs the Compilers Creating Custom Processors research group. His research interests include technologies of application-specific processors and accelerators, resilient computer systems, and compilers. He has a BS in computer engineering, and an MS and a PhD in electrical engineering, all from the University of Illinois at Urbana-Champaign. He is a member of the IEEE and the ACM.

Yu Cao is an assistant professor at Arizona State University, Tempe. His research interests include physical modeling of nanoscale technologies, design solutions for variability and reliability, and reliable integration of post-silicon technologies. Cao has a BS in physics from Peking University, an MA in biophysics from the University of California, Berkeley, and a PhD in electrical engineering from the University of California, Berkeley. He is a member of the IEEE and the ACM.

■ Direct questions and comments about this article to Todd Austin, 4637 CSE-ACAL Lab, University of Michigan, 2260 Hayward, Ann Arbor, MI 48109-2121; austin@umich.edu.

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

The Changing Design Landscape

Ajith Amerasekera, Texas Instruments

In 1965, Gordon Moore introduced his now famous law on the doubling of IC complexity. For many decades, this doubling has been realized through process technology innovation that has enabled semiconductors to achieve their complexity roadmap through scaling of transistor geometries. As we approach the 45-nm technology node of the *International Technology Roadmap for Semiconductors* (*ITRS*), this scaling is alive and continuing, but some of the supporting infrastructure is not tracking it. In particular, performance is no longer increasing by significant amounts at every node. In conjunction with this, the equipment capabilities to print these incredibly small geometries are now the main driver for the process technology.

Moore's law still continues, and will do so as long as our system requirements expand at the existing rate. However, the burden of enabling Moore's law to continue is gradually moving from the process technologists to the designers—both circuit and system designers. Every two years or so, the increase in available capacity for integration exceeds the capability developed since Moore first proposed his law. For example, from 65 nm to 45 nm, the number of logic gates per square millimeter increased from 700,000 to 1.4 million. Considering that the Intel 486 processor, one of the technology drivers of the desktop computing era, had about 1 million transistors, this one process node provided an increase in capacity equivalent to an entire CPU, and at a far higher performance than was available back in the 1990s.

Table 1 shows what the scaling factors will be as we move from 65 nm to 22 nm. The number of transistors on a big chip will go from approximately 2 billion in 65 nm to 9 billion in 28 nm, to 15 billion in 22 nm. These are all huge numbers. The challenge is to build these chips with the same robustness, yield, and reliability as previous generations of chips.

The techniques described in "Reliable Systems on Unreliable Fabrics" (by Todd Austin et al.) in this issue of *IEEE Design & Test* describe some of the design issues related to this huge increase in complexity. In particular, technology scaling and the manufacturing process come with higher variations in transistors both locally and globally on a chip. Designers must account for this variation. Statistically, the large number of components on a single chip will lead to reliability, aging, and defect limitations that could no longer be eliminated through margins or overdesign. They must be detected and compensated without affecting the performance goals of the chip.

The research direction of the GSRC is aimed at solutions to these obstacles, for the continued advancement of system performance needs. Concepts such as error resiliency and fault-tolerant design are the first steps toward enabling chip designers to take control of their roadmap. Adaptive architectures and adaptive design will take advantage of the billions of transistors available in these nodes to provide techniques for ensuring robustness in these huge designs.

This is just the beginning of the work that must be done to build the large SoC designs with multibillion transistors that will be commonplace below the 32-nm technology node and beyond. GSRC researchers working with industry leaders will ensure that Moore's law continues to drive the electronics industry and the consumer world for many decades to come.

Ajith Amerasekera *is a TI Fellow and ASIC CTO at Texas Instruments. Contact him at amerasekera@ti.com.*

Table 1. Some typical transistor and gate metrics as technology scales from 65 nm to 22 nm.

	***ITRS* process technology node**					
Metric	**65 nm**	**45 nm**	**40 nm**	**32 nm**	**28 nm**	**22 nm**
Scale (from 65 nm)	1	69%	62%	49%	43%	34%
Route pitch	210	140	126	100	90	70
Raw Mgates/mm^2 on silicon	0.70	1.47	1.81	2.83	3.70	5.99
25 × 25 die gates (billions)	0.440	0.918	1.133	1.771	2.313	3.747
25 × 25 die transistors (billions)	1.8	3.7	4.5	7.1	9.3	15.0
Typical chip design (Mgates)	51	106	130	204	266	431

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