Vision for Cross-Layer Optimization to Address the Dual Challenges of Energy and Reliability

(Invited Paper)

André DeHon
Electrical and Systems Engineering
University of Pennsylvania
200 S. 33rd St.
Philadelphia, Pennsylvania 19104
Email: andre@acm.org

Heather M. Quinn
Los Alamos National Laboratory
ISR-3 Space Data Systems
Los Alamos, NM 87545
Email: hquinn@lanl.gov

Nicholas P. Carter
Intel Corporation
2200 Mission College Blvd, RNB6-61
Santa Clara, California 95054
Email: nicholas.p.carter@intel.com

***Abstract*—We are rapidly approaching an inflection point where the conventional target of producing perfect, identical transistors that operate without upset can no longer be maintained while continuing to reduce the energy per operation. With power requirements already limiting chip performance, continuing to demand perfect, upset-free transistors would mean the end of scaling benefits. The big challenges in device variability and reliability are driven by uncommon tails in distributions, infrequent upsets, one-size-fits-all technology requirements, and a lack of information about the context of each operation. Solutions co-designed across traditional layer boundaries in our system stack can change the game, allowing architecture and software (a) to compensate for uncommon variation, environments, and events, (b) to pass down invariants and requirements for the computation, and (c) to monitor the health of collections of devices. Cross-layer codesign provides a path to continue extracting benefits from further scaled technologies despite the fact that they may be less predictable and more variable. While some limited multi-layer mitigation strategies do exist, to move forward redefining traditional layer abstractions and developing a framework that facilitates cross-layer collaboration is necessary.**

I. Introduction

For over 40 years we have been able to rely on fabrication improvements to produce high yielding integrated circuits that have adequate noise and upset tolerance that satisfy most applications. Reliability of devices has been a manufacturing problem—we simply demand that manufacturing continue to produce devices with adequately high yield and upset tolerance. These properties have continued to hold across scaling. While pre-integrated circuit work did contemplate the need to deal with highly error-prone devices [1], current semiconductor circuit designers, computer architects, and firmware and software developers have largely been able to assume perfect operation from the devices. DRAM memory bits are a notable exception to this rule, and the techniques they employ hint at radically different approaches we might take to achieving reliability.

At the circuit level, we make a single concession to device variability due to process, environment, aging, and noise—margins. To accommodate the large spread in device characteristics that these variability effects induce, we operate our circuits with large timing and voltage margins around their nominal characteristics. For example, by selecting the voltages large enough, we assure that our circuits continue to function correctly despite any variation in device threshold voltage. This margining is necessary, since a non-trivial fraction of the fabricated devices on a modern integrated circuit will have characteristics far from the intended target. We also add additional margins to the supply voltage to accommodate a wide range of temperatures and potential supply noise. In this way we spend energy—the extra voltage margin to guarantee correct operation across all likely variation cases and environment scenarios—to combat a potential reliability problem.

Unfortunately, we are now approaching a convergence of two inflection points.

1) Energy – limits on practical power dissipation has now reached a point where energy concerns (both power density and absolute energy draw) limit the computation we can deploy on a chip. The primary driver in computational design shifts from transistor density and speed to power density and energy cost [2], [3].
2) Reliability – variation in parameters due to small scale effects coupled with larger device counts are rapidly driving the need for higher percentage margins. While the mean energy and delay may continue to decrease with scaling, the expected worst-case devices on the chip could have higher delay and demand higher voltages for correct operation.

This convergence presents a challenge to our status quo approach to reliability. Our need to continue to reduce energy per device operation to increase the performance delivered per Joule or per W/cm^2 is limited by our need to provide increasing margins to deal with more variable and noisy devices, threatening an end to beneficial scaling. While it is possible to continue to produce smaller feature size components, following the traditional approach of using energy margins to hide reliability effects at the circuit level will prevent further reduction in the energy per device operation.

If we are to continue scaling, we must dramatically change how computing systems are designed. Rather than demanding

perfectly manufactured devices that do not change over their lifetimes and work for all environments, we must permit devices to fail and compensate for their failure at higher levels in our system stack. Rather than making reliability solely the responsibility of manufacturing, reliability management becomes a cooperative effort across the system stack involving circuit design, architecture, firmware, operating systems, middleware, compilers, and application software. In particular, since it is the uncommon events (*e.g.*, tails of the process parameter distribution, infrequent upsets) that drive reliability problems, cross-layer solutions that only spend extra energy to handle these exceptional events will be more economical than circuit-layer margining that charges all devices and operations the large energy tax necessary to guarantee correct operation for the uncommon devices and events. This situation suggests a cross-layer, full-system co-design approach to efficiently compensate for the new reliability challenge.

II. Trends

Several scaling trends converge to exacerbate the challenge ahead: increasing device and component counts, increasing variability, increasing burnout and wear, decreasing voltage, and increasing deployment of integrated circuits into critical roles.

Flat power density budgets, such as 100W/cm^2 for forced-air cooling or 1–10W/cm^2 for ambient cooling, coupled with increasing transistor count and slowly reducing capacitance, demands voltages scale down with feature sizes. However, since transistor subthreshold slope does not scale and we need to maintain high I_{on}/I_{off} ratios, we cannot scale voltage down aggressively enough to meet the power density limit if all devices switch. Limited voltage scaling leads to the current inflection where the power density budget prevent us from activating all the devices we can potentially manufacture on a circuit. Nonetheless, *absent reliability concerns,* it remains possible to reduce the absolute energy required per switched device.

Decreasing feature size leads to increasing variation as noted in both the ITRS [4] and the companion resilience roadmap article [5]. Conventional margining techniques set operating voltages to guarantee correct operation across the expected range of devices characteristics (*e.g.* $\pm 3\sigma$). When the standard deviation becomes a significant fraction of nominal voltage (*e.g.* $\sigma_{V_t}/V_t \rightarrow 27\%$ before F=22nm [4]), an increasing percentage of the voltage swing must be dedicated to margining for worst-case devices. This effect limits or reverses voltage scaling for fixed yield goals. Alternately, continued voltage scaling means increased defect rates as shown in the companion resilience roadmap. [6] shows an example where the minimum energy per operation considering the expected variation actually increases as we scale from the 45nm to the 32nm node.

At the same time, we continue to increase the number of transistors per integrated circuit, increasing the number of transistors that are statistically sampling from the device parameter distribution. To achieve comparable chip-level yields via margining, this forces us to accept a larger spread of device characteristics. That is, if we needed $\pm 3\sigma$ margins to get adequate yield at smaller transistors counts, we might be forced to now tolerate $\pm 4\sigma$ margins. The alternative is to expect a larger number of intolerably bad devices on each component.

Decreased feature size and voltages also mean a decrease in the critical charge holding state, increasing upset susceptibility. At the same time, we are placing more transistors on a chip and more components in large-scale systems such as supercomputers [7] and data centers [8]. These state-of-the art large-scale systems will see a composite increase in upset rates from these two effects. Unmitigated, these effects decrease the mean-time-to-system-failure.

Finally, decreased opportunities for device burnin [9] mean more weak devices will escape initial test and fail in the field. Increasing wearout effects, including negative-bias temperature instability [10] and hot carrier injection [11], further expand the spread in component characteristics, demanding even greater margins using traditional solutions, or increasing the rate of field failure and decreasing component lifetime [12].

These reliability challenges come at a time when the impact of failure is increasing. Electronics are being deployed more pervasively into all aspects of our lives (*e.g.* cell phones, PDAs, business transactions), into our critical infrastructure (*e.g.* building, power grid, financial, e-commerce, communications, GPS satellites), and into life critical roles (*e.g.* automotive, aerospace, medical components). Our modern world increasingly depends on the reliable operation of a growing number of these devices, both increasing the susceptibility and impact of integrated circuit failure. This situation drives an increasing need for higher reliability systems—a trend opposite of where device-level scaling is headed. The result is a widening gap between device-level reliability and system-level reliability requirements.

III. Old Solutions

Traditionally, we have demanded that all the logic devices on a chip yield, discarding the integrated circuit when any transistor fails to operate correctly. As noted above, we employ margins to tolerate varying device characteristics. It has been the job of manufacturing to guarantee parametric device yield rates are high enough to guarantee chip-level yield. We further rely on energy margins to guarantee that the probability of state and logic upset is sufficiently low that we can assume the data is never corrupted. Bulk storage in large memories (DRAMs) and persistent storage (hard and solid-state disks) represent a notable exception where we tolerate both errors in manufacture and errors in operational state using architected redundancy and efficient error detection and correction coding.

In those cases where we demand higher system-level reliability than manufacturing happens to provide, we have relied on brute-force replication of components. For example, aeronautic and space systems often run three or more computations in parallel and vote on the result to avoid single or multiple failures [13], [14]. Critical commercial systems run two copies

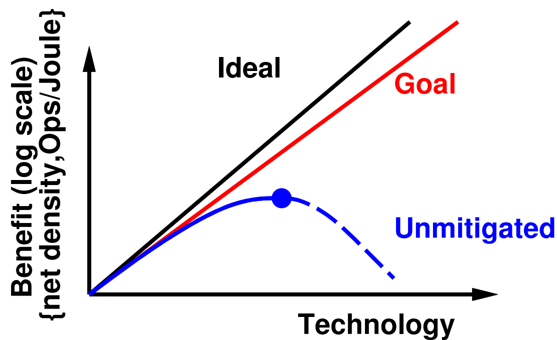

Fig. 1. Scaling Scenarios

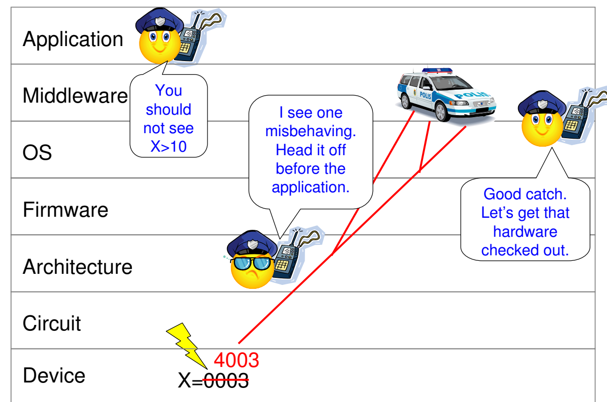

Fig. 2. Cartoon Illustration of Cross-Layer Cooperation

in parallel to detect errors [15], [16]. Distributed systems use redundant servers to balance the load and tolerate both interconnect and node outages, relying on the ability to obtain sufficiently equivalent service from different resources.

IV. End of Beneficial Scaling

The traditional benefit of scaling has been the decrease in cost-per-user-visible functionality. This benefit comes from technological effects, such as a decreasing cost per gate and decreasing energy per gate evaluation. If increasing margins means an increase in energy per gate, the new, scaled technology offers no advantage over the previous technology, as shown in the "Unmitigated" curve in Figure 1. Similarly, if mitigating reliability problems means triplicating logic and voting, the scaled technology might not offer a reduction either of energy or area. The net result of mitigating the reliability problem will be an increase in area and energy per gate. Both effects suggest it will not be economically beneficial to use the scaled technology. Consequently, we must find more economical ways to enhance system reliability above the device level to continue to exploit the benefits of further feature-size scaling.

V. Goal

Our goal is to facilitate the successful navigation of the energy and reliability inflection points. Specifically, this means finding solutions that maintain or improving system safety while allowing continued scaling benefits. To continue scaling, we must continue to deliver increased operations per time while working within a fixed power-density budget. To achieve this end, we accept that raw device reliability and consistency will decrease and look for ways to build reliable and predictable systems from unreliable and unpredictable devices. Modern energy and power challenges demand that the mitigation techniques used to compensate for unpredictable devices be energy efficient. That is, they must require only a small energy investment and lead to net energy reductions relative to unscaled solutions, as shown in the "Goal" curve in Figure 1.

We believe cross-layer cooperation from higher levels of the system stack, as illustrated in Figure 2, is essential to achieving the necessary efficiency. We are beginning to see scattered solutions with this flavor. Nonetheless, this approach demands a wholesale paradigm shift in the way we design and engineer computer systems.

VI. Cross-Layer Information Flow Motivation

We can trace the root causes of many of the challenges and compromises seen in today's system designs to the need to design and operate specific system layers without key information. This underscores the potential opportunities for cross-layer information sharing to address power and reliability challenges. In this section, we identify several such information deficiencies as partial motivation for the vision that follows.

a) Late-Bound Information: Environment, energy demands, deployed system context, and even technology noise and maturity are all late bound, often unknown during design and perhaps not known until deployment. The lack of this information leads to both over-design for most scenarios and limited ability to use the components in more demanding scenarios (*e.g.* higher defect and fault rate than anticipated, larger environmental variations, more critical deployment contexts). This situation motivates the design of components and systems with modes and configuration options that allow higher layers in the system to tune what the components spends on reliability once this information becomes known. These modes will allow commercial devices to enhance yield or operate at extremely low energy levels [17] while also making the same parts more usable in larger scale systems or harsher environments.

b) Instantaneous Operational Information: Varying demands, workloads, environment and uncertainty about the environment even in a single system means margins and mitigating techniques designed for the worst-case possible scenario are over-design for most operating hours [18]. This problem suggests a need for systems that can monitor their environment and health to extract the missing information and adapt to exploit this information about their situational needs.

c) Information about Application Requirements: Worst-case design for a platform independent of application needs is too expensive. This overdesign arise when we demand that the

platform, such as a supercomputer, provide a fixed, minimum level of reliability with no information about the tasks that are running on it. Similarly, worst-case design for uncommon, but potentially avoidable, worst-case scenarios is also a large, unnecessary cost [19]. This challenge motivates cross-layer, application-aware solutions. These solutions may include models and middleware that allow the application to communicate requirements and opportunities to the platform and that support management of operational and implementation aspects of an application mapping to a particular platform.

d) Information about Capabilities and Health of Components from Heterogeneous Suppliers: Fully custom or unique construction of all components is not viable for any company, industry, or government agency. Some domains see more acute versions of this problem, but no domain can afford the investment in time, manpower, and unique manufacturing costs to develop all components custom for their applications and systems. This scenario motivate the need for interfaces, metrics, benchmarks, and tools to perform composition, analysis, optimization, and validation of separately sourced (sub)components. System solutions must be cross-layer, with higher-layers conveying context to lower layers and lower-layers communicating capabilities and health to upper layers. Lack of information drives inefficient and conservative use of components.

e) Incomplete Information on Component and System Reliability Weaknesses: Across the board, there is considerable conservative overdesign. Time-to-market pressures and limits on human time coupled with a lack of automation drives the acceptance of large margins and safety factors at many layers of the design. In many cases the safety factors are not effectively applied—providing too much guarding on most cases and components in order to get an adequate level on a subset. This problem arises from a lack of visibility into the real sources of weakness in the design. There is a need for system assessment methodologies and tools to support better and more automated exploration of tradeoffs in the energy-delay-area-reliability-thermal-mechanical design space. This situation is true both for chip-level design of processors and ASICs and for system-level design of satellites, supercomputers, cell-phones, and pacemakers.

VII. Vision

We can no longer assume computational elements will be perfectly and identically fabricated and operate without transient upsets. While multi-level solutions have been useful in protecting bulk storage, such as DRAMs, persistent storage, similar solutions for computation currently do not exist. In part, the heterogeneous design of computing systems and their ability to transform data makes posing simple solutions that do not rely on brute-force replication difficult. Nonetheless, there are hints of abundant opportunities for economical cross-layer protection of computations.

Hardware organizations must be prepared for repair. Both RAM and hard disks expect errors and employ microarchitectures and abstractions that allow repair. While RAM repair, such as row and column sparing, occurs below the architectural level and is invisible to the software, bad disk sectors in hard drives are visible to the operating system. In a similar manner, our computational organizations must be prepared for errors. Reconfiguration provides the ability to exploit late-bound information about where high variation or defects have occurred so they can be avoided or allocated where their impact is beneficial [20], [21]. Mitigation of the errors will likely require cooperation across the microarchitecture, architecture, and operating system.

Errors must be filtered at multiple levels. To use small devices, memory systems allow individual memory bits to fail. The microarchitecture assists by correcting errors during memory access. The operating system collaborates by scrubbing memory. To use small devices and low energy for computation, we must similarly expect occasional errors in the computation. These errors will need to be caught and corrected at higher levels in the system stack.

Multilevel trade-offs provide efficient solutions, generalizing the idea of hardware-software trade-offs. With errors slipping through devices, higher levels must be prepared to detect and correct the errors. Similar to the way we distribute the function of virtual address translation across hardware (*e.g.* translation lookaside buffer) and software (*e.g.* miss handling and replacement), efficient solutions will carefully divide functionality between the microarchitecture and system software. This solution avoids paying a large energy costs for uncommon events. Suitable architectural interfaces will be required and will benefit from compiler and application support.

Strategic redundancy improves solution efficiency. Information theory tells us how to provide shared redundancy across large blocks of data to avoid brute-force replication. Efficient computational solutions will similarly avoid brute-force replication. For example, invariants and end-to-end consistency checks on the computation may allow for lightweight checks of errors. Characterization of the origin and reproducibility of data may allow more efficient state protection and checkpointing [22]. This further allows the hardware to safely operate on the edge of failure, using information to detect and recover when the system goes over the edge, avoiding the need to spend energy in margins to guarantee the edge is never encountered [23].

Differential reliability enables more efficient solutions. DRAMs with Error-Correcting Codes (ECC) and row sparing carefully exploit the fact that the ECC allows the core of the memory to be less reliable than the periphery. This solution also exploits the ability to fabricate devices with different feature sizes to assure stronger reliability. Computations can similarly employ a mix of larger, more-reliable devices and smaller, less-reliable devices. Similarly, we can use higher voltages and currents to make some circuits more reliable than others. Thereby, computations that have efficient checks or are less sensitive to errors can be run on smaller, lower-energy devices. In this manner, high-level information about application invariants or requirements drives microarchitectural decisions

around the deployment of circuits and devices with different characteristics.

As a multi-level cache memory system attempts to provide the density of a large memory with the speed of small memory:

- A traditional, ECC-protected memory provides the reliability of large feature sizes with the density of small memory cells.
- Multi-level computational designs can provide the reliability of large-feature and large-energy devices with the density and energy consumption of small-feature, low-energy devices.

Scalable solutions should allow adaptation to error rates and reliability. Scalability to different error rates and different levels of protection is not present in traditional DRAM memory systems. Nonetheless, information theory does tell us how to develop codes of different rates to handle different needs, and it is easy to see how to add adaptability for memory systems. With growing error rates, error rates that vary with environment, and applications with differing needs for protection, we need the engineering understanding of how to best provide that protection across the design space as well as architectures and components that can be tuned in-system to varying environmental conditions. Device wear suggests error rates will change over time in a single component, further driving the need for in-system adaptation.

Components should degrade gracefully and the system should be aware of its overall health. The system should not move from a state of correct operation to one of failure without noticing early-warning signs. It should be able to assess its readiness before performing tasks and self-report when it cannot meet the requested level of reliability.

VIII. Impact

Addressing these issues are essential to safety and the world-wide economy.

Economic: The growth of the world-wide economy and well being has been fueled by cheaper and more powerful computations that enable greater automation and new services and products. This growth, in turn, has been fueled by Moore's Law scaling. Integrating reliability and variation management into designs is essential to allowing us to continue to extract size, cost, and energy benefits from scaled computations.

Energy: Energy consumption promises to be a limitation to our capabilities, our economy, and our environmental impact. Continued reduction in the amount of energy consumed per computation remains an important tool in expanding our computing capacity and taming our energy consumption.

Ultra-reliable Systems: Computation increases our infrastructural capabilities and our efficient use of scarce resources. Unless we systematically address reliability issues, these systems will be hit by the double-whammy of increasing device count and decreasing device reliability.

Harsh Environments: Automated computations expand our reach and survivability into harsh environments, such as space, high altitude, or extreme temperatures. However, these environments increase the upset and wear rates for devices, an effect that is further magnified as devices scale down in size. We must be able to scale our reliability solutions to these more extreme environmental characteristics and do so with modest incremental effort on top of mainstream designs.

Security: As more of our interactions are managed and enhanced by computer mediation, it becomes increasingly critical that these systems be robust against deliberate subversion attempts. It is a difficult task to guarantee that a system cannot be penetrated even when we assume the devices and components work perfectly. Misbehaving devices violate key assumptions and create a myriad of new attack vectors against our systems. For example, researchers have already identified ways in which soft errors can be used to defeat cryptographic systems [24], [25] and software isolation layers [26].

IX. Platforms and Education

Flexibility and adaptability are one of the strengths of modern computing systems. The mark of a successful computing platform or tool is that they are regularly deployed into uses beyond those originally envisioned by the designer. Much of the economic benefit for building on common, standard platforms is the effort reduction and cost savings associated with avoiding the need to build a new system from scratch. Between both the large fraction of computing systems that are employed in critical roles and the potential for almost any system to be deployed into such a role, modern platforms must be designed with scalable noise-tolerance, reliability management, and adaptation in mind.

The computer engineer can no longer assume that device manufacturing delivers adequate reliability. Consequently, he or she must address reliability as a design goal along with energy, area, and delay. Furthermore, as a larger fraction of computing systems are or may be deployed into critical roles, the safety and survivability of our highly automated modern world depends on active reliability management by the computer engineers. Just as safety management is concern for all civil engineers, cross-layer reliability management must become a concern for all computer engineers. Computer engineering education must evolve rapidly to prepare engineers for the new reality.

X. Conclusion

Continued feature size scaling brings the convergence of inflection points in energy and reliability. These situations lead to conflicting demands on voltage scaling that suggest the end of beneficial scaling if conventional approaches to reliability and energy management are retained. Continued scaling while maintaining or increasing system-level reliability demands a paradigm shift away from relying on device-level reliability and toward cooperative, cross-layer reliability management. Key to these cross-layer solutions is strategic sharing of information between layers and continual exploitation of information to adapt throughout operation. Traditional layer interfaces and contracts must be redefined to facilitate this cooperation. Embracing these changes will allow us to address a number of emerging areas of pain across the industry.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 0637190 to the Computing Research Association. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Computing Research Association, the National Science Foundation, or Intel Corporation. This vision was developed as part of a Computing Community Consortium (CCC) visioning study to develop and build community consensus on emerging challenges and identify opportunities and priorities for research to address them. The study brought together over 80 industry, academic, and government engineers to discuss these issues. The vision articulated here builds on insights and input from the workshop participants. For a full list of participants and further information on the study, visit <http://www.relxlayer.org>. Document release number: LA-UR-09-07761.

References

[1] J. V. Neumann, "Probabilistic logic and the synthesis of reliable organisms from unreliable components," in *Automata Studies*, C. Shannon and J. McCarthy, Eds. Princeton University Press, 1956.

[2] M. Horowitz, E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein, "Scaling, power, and the future of CMOS," in *Technical Digest of the IEEE International Electron Device Meeting*, December 2005, pp. 7–15.

[3] B. Nikolic, "Design in the power-limited scaling regime," *IEEE Transactions on Electron Devices*, vol. 55, no. 1, pp. 71–83, Nanuary 2008.

[4] "International technology roadmap for semiconductors," http://www.itrs.net/Links/2008ITRS/Home2008.htm, 2008.

[5] S. R. Nassif, N. Mehta, and Y. Cao, "A resilience roadmap," in *Proceedings of the Conference and Exhibition on Design, Automation and Test in Europe*, 2010.

[6] D. Bol, R. Ambroise, D. Flandre, and J.-D. Legat, "Interests and limitations of technology scaling for subthreshold logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 10, pp. 1508–1519, 2009.

[7] K. J. Barker, K. Davis, A. Hoisie, D. J. Kerbyson, M. Lang, S. Pakin, and J. C. Sancho, "Entering the petaflop era: the architecture and performance of roadrunner," in *Proceedings ACM International Conference on Supercomputing*, 2008, pp. 1–11.

[8] L. A. Barroso and U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, ser. Synthesis Lectures on Computer Architecture. Morgan & Claypool, 2009, no. 6.

[9] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10–16, November–December 2005.

[10] D. K. Schroder and J. A. Babcock, "Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing," *Journal of Applied Physics*, vol. 94, no. 1, pp. 1–18, July 2003.

[11] S.-H. Renn, C. Raynaud, J.-L. Pelloie, and F. Balestra, "A thorough investigation of the degradation induced by hot-carrier injection in deep submicron n- and p-channel partially and fully depleted unibond and SIMOX MOSFETs," *IEEE Transactions on Electron Devices*, vol. 45, no. 10, pp. 2146–2152, October 1998.

[12] L. Condra, J. Qin, and J. Bernstein, "State of the art semiconductor devices in future aerospace systems," in *Proceedings of the FAA/NASA/DoD Joint Council on Aging Aircraft Conference*, April 2007.

[13] Y. C. B. Yeh, "Triple-triple redundant 777 primary flight computer," in *Proceedings of the Aerospace Applications Conference*, 1996, pp. 293–307.

[14] R. E. Lyons and W. Vandekulk, "The use of triple-modular redundancy to improve computer reliability," *IBM Journalof Research Development*, vol. 6, no. 2, p. 200, 1962.

[15] D. E. Lenoski, "A highly integrated, fault-tolerant minicomputer: The nonstop CLX," in *Digest of Papers—Compcon Spring 88: Intellectual Leverage*. IEEE, February 1988, pp. 515–519.

[16] T. Slegel, I. Averill, R.M., M. Check, B. Giamei, B. Krumm, C. Krygowski, W. Li, J. Liptay, J. MacDougall, T. McPherson, J. Navarro, E. Schwarz, K. Shum, and C. Webb, "IBM's S/390 G5 microprocessor design," *IEEE Micro*, vol. 19, no. 2, pp. 12–23, Mar/Apr 1999.

[17] C. Wilkerson, H. Gao, A. Alameldeen, Z. Chishti, M. Khellah, and S.-L. Lu, "Trading off cache capacity for reliability to enable low voltage operation," in *Proceedings of the International Symposium on Computer Architecture*, June 2008, pp. 203–214.

[18] M. Caffrey, K. Morgan, D. Roussel-Dupre, S. Robinson, A. Nelson, A. Salazar, M. Wirthlin, W. Howes, and D. Richins, "On-orbit flight results from the reconfigurable cibola flight experiment satellite (CFESat)," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, 2009, pp. 3–10.

[19] V. J. Reddi, M. S. Gupta, G. Holloway, M. D. Smith, G.-Y. Wei, and D. Brooks, "Voltage emergency prediction: A signature-based approach to reducing voltage emergencies," in *Proceedings of the International Symposium on High-Performance Computer Architecture*, 2009.

[20] A. DeHon and H. Naeimi, "Seven Strategies for Tolerating Highly Defective Fabrication," *IEEE Design and Test of Computers*, vol. 22, no. 4, pp. 306–315, July–August 2005.

[21] B. Gojman and A. DeHon, "VMATCH: Using Logical Variation to Counteract Physical Variation in Bottom-Up, Nanoscale Systems," in *Proceedings of the International Conference on Field-Programmable Technology*. IEEE, December 2009.

[22] J. N. Glosli, K. J. Caspersen, J. A. Gunnels, D. F. Richards, R. E. Rudd, and F. H. Streitz, "Extending stability beyond CPU millennium: A micron-scale atomistic simulation of kelvin-helmholtz instability," in *Proceedings ACM International Conference on Supercomputing*, 2007.

[23] T. Austin, D. Blaauw, T. Mudge, and K. Flautner, "Making typical silicon matter with Razor," *IEEE Computer*, vol. 37, no. 3, pp. 57–65, March 2004.

[24] J. Xu, S. Chen, Z. Kalbarczyk, and R. K. Iyer, "An experimental study of security vulnerabilities caused by errors," in *Proceedings of International Conference on Dependable Systems and Networks*, 2001, pp. 421–432.

[25] A. Shamir, "Research announcement: Microprocessor bugs can be security disasters," Available online at http://cryptome.org/bug-attack.htm, November 2007.

[26] S. Govindavajhala and A. W. Appel, "Using memory errors to attack a virtual machine," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2003.