Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation

As technology scales, variability in transistor performance will continue to increase, making transistors less and less reliable. This creates several challenges in building reliable systems, from the unpredictibility of delay to increasing leakage current. Finding solutions to these challenges will require a concerted effort on the part of all the players in a system design.

Shekhar Borkar
Intel Corp.

VLSI performance has increased by five orders of magnitude in the last three decades, made possible by continued technology scaling. This trend will continue, providing an integration capacity of billions of transistors; however, power, energy, variability, and reliability are barriers to future scaling.

Die size, chip yields, and design productivity have so far limited transistor integration in a VLSI design. Now the focus has shifted to energy consumption, power dissipation, and power delivery.[1] Transistor subthreshold leakage continues to increase, and those of us in this industry have devised leakage avoidance, tolerance, and control techniques for circuits.[2] As technology scales further we will face new challenges, such as variability,[3] single-event upsets (soft errors), and device (transistor performance) degradation—these effects manifesting as inherent unreliability of the components, posing design and test challenges. This article discusses these effects and proposes microarchitecture, circuit, and testing research that focuses on designing with many unreliable components (transistors) to yield reliable system designs.

This problem is not new; we design systems to account for reliability issues. For example, error-correcting codes are commonly used in memories to detect and correct soft errors. Careful designing and testing for frequency binning copes with variability in transistor performance. What is new is that as technology scaling continues, the impact of these issues increases, and we need to devise techniques to effectively deal with them.

Sources of variations

There are three major sources that cause variations in transistor behavior. The first source is

Published by the IEEE Computer Society

called random dopant fluctuations, which results from discreteness of dopant atoms in the channel of a transistor.[4] Transistor channels are doped with dopant atoms to control their threshold voltage. Figure 1 shows dopant atoms in the channel of several generations of transistors. The decrease in transistor size each technology generation reduces its area by half, and thus the number of dopant atoms in the channel decreases exponentially over generations. Notice that in the 1-micron technology generation there were thousands of dopant atoms, whereas in a 32- to 16-nm generation there are only tens of dopant atoms left in the channel, and the law of large numbers no longer applies. Therefore, two transistors sitting side by side will have different electrical characteristics because of randomness in a few dopant atoms, resulting in variability.

The second source of variability is because of sub-wavelength lithography. Since the 0.25-µm technology generation, we have used sub-wavelength lithography for patterning transistors. For example, fabrication processes used a 248-nm wavelength of light to pattern 0.25-µm (250-nm) and 0.18-µm transistors. The wavelength decreased to 193 nm for 130-nm technology and has since remained constant for even 65-nm transistors. There might be some additional breakthroughs to effectively reduce this wavelength (to a 157-nm light source or via immersion technology) but the difference in the wavelength of light and the patterning width will continue to widen until extreme ultra-violet technology (13 nm) becomes available. Until then, sub-wavelength lithography is here to stay. This sub-wavelength lithography is the primary reason for line edge roughness and several other effects in transistors, resulting in variations.

These first two sources are static—that is, they occur during fabrication—but the third source of variations is dynamic; that is, it is time and context variant. Figure 2 shows heat flux (power density) in Watts per square centimeter across a microprocessor die. The heat flux across the die varies depending on the functionality of the circuit block. For example, a cache has less heat flux than an execution unit, and it depends on the activity and compute load at any given time. Higher heat flux also puts more demand on the power distribution grid, resulting in resistive and inductive voltage drops, and creating time-dependant, dynamic, supply voltage variations. Higher heat flux results in higher temperature, creating hot spots, which in turn create temperature variations across the die, affecting circuit performance. This also results in higher subthreshold leakage, variations in the leakage across the die, and variations in power delivery demand across the power distribution grid. Supply voltage variations will have a worse impact in future technology generations as the supply voltage scales (even moderately) and the supply current increases.

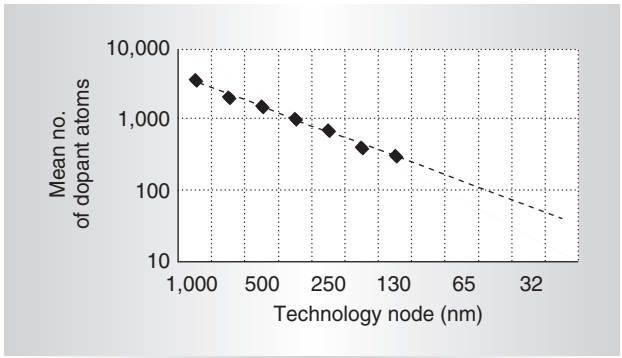

Figure 1. Random dopant fluctuations.

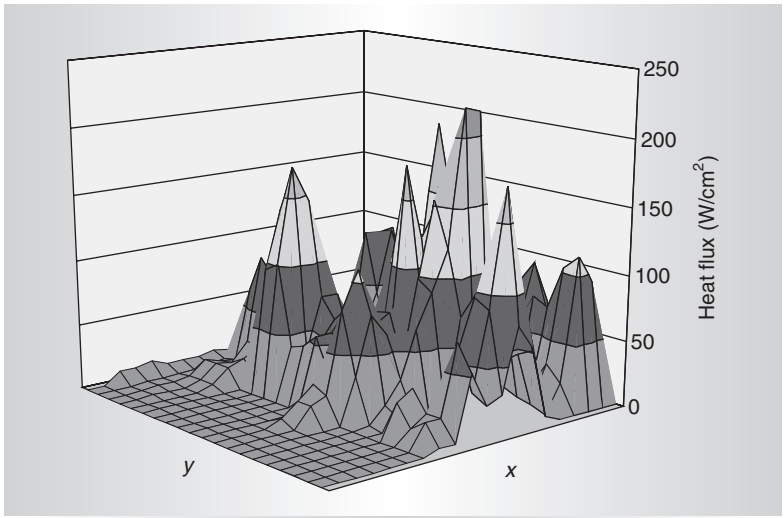

Figure 2. Heat flux.

Impact of variations

Static and dynamic variations have impact even on today's VLSI chips, and it's expected to get worse as technology scales.[3] Today, we see the effect predominantly as a 30 percent variation in operating frequency and 5 to 10

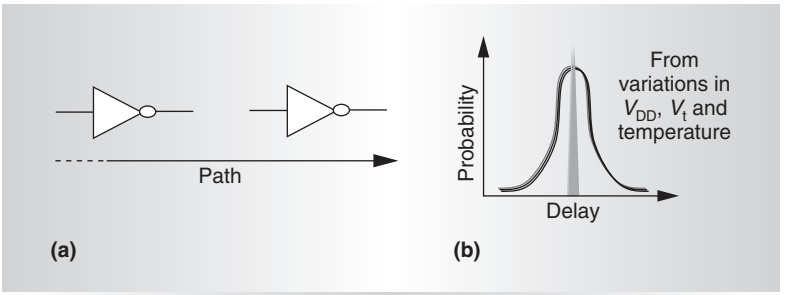

Figure 3. Deterministic (a) versus probabilistic (b) path delay.

times variation in leakage power. Since subthreshold leakage power is a major portion (30 to 50 percent) of total power consumption, a 5 to 10 times variation in the leakage power alone contributes to almost a 50 percent variation in total power. Since the behavior of the fabricated design in terms of power and performance differs from what designers intended, the effect of variations looks like inherent unreliability in the design.

Variation-tolerant design

Researchers have proposed numerous process technology, circuit, and architectural solutions to combat variations;[3,5,6] these solutions require radical changes in design methodology. For example, designers can use forward and reverse body bias to tighten subthreshold leakage and frequency distributions. Chips with higher leakage tend to be faster, hence it is possible to apply reverse body bias to reduce leakage and reduce frequency. Similarly, slow chips can benefit from forward body bias to improve their speed at the expense of a moderate increase in leakage power. Similarly, adaptive supply voltage, used in conjunction with body bias can tighten the distribution.[5]

Chip frequency depends on the speed of critical paths. During circuit design, designers model a critical path, Figure 3a, to have deterministic delay as predicted by a circuit simulator. However, because of the static and dynamic variations just described, the circuit's delay, illustrated in Figure 3b, is probabilistic. When designers complete the design and apply conventional design methods, they typically downsize transistors to save active power. As a result, they also downsize transistors in the paths close to critical paths, increasing the total number of critical paths in the design, hence exasperating the impact of variations.

Although manufacturers make every attempt to maintain the deterministic behavior, the increased variability—or not fully comprehended variability in transistor performance—can make these path delays probabilistic. Therefore, the industry must deviate from the conventional methodology of downsizing transistors indiscriminately to reduce active power because downsizing indiscriminately makes many noncritical paths critical, and reduces the probability of meeting the frequency goal. Similarly, low-threshold-voltage transistor usage does not have to be minimal to reduce leakage power. With reduced low-V_t usage across the design, the transistors near the critical path might also be replaced with high-V_t transistors, and because of variations in V_t these paths could become slower, resulting in a wider frequency distribution. Design tools and methodologies must comprehend variations and optimize the design not for frequency alone, but for active and leakage powers, and their distribution.

When we design a microarchitecture, the tendency is to improve the frequency of operation by creating more critical paths, which reduces the probability of meeting the increased frequency goal. Furthermore, to meet higher frequency goals, the microarchitecture tends to employ fewer gate delays in a clock cycle. Since passing the signal through fewer gates in a clock cycle does a poor job of averaging and canceling the effects of variations, there is a lower probability of meeting the frequency goal. This, once again, is contrary to conventional thinking and design methodology.

We need to evolve from today's deterministic design to probabilistic and statistical design for the future. These new design methods must account for variations, and optimize for yield, performance, and power.

Extreme variations

As technology continues to scale further—beyond 22 nm or so—both static and dynamic variations will continue to become worse for the reasons discussed, resulting in wider distribution of transistor threshold voltages. Figure 4 illustrates this trend. In Figure 4, the target V_t of the transistor is 150 mV. In current process technologies, the measured V_ts of the actual devices tend to cluster tightly

around the target V_t. In the future (the dashed line), for nanometer-scale technologies, fewer transistors will actually approach the target V_t, and this distribution will flatten out. These variations in the transistors could become severe enough that it would be impossible to correct for them during design—you might have to somehow compensate for them at the level of the entire system.

Single event upsets (soft errors) are another source of concern. These errors are caused by alpha particles and, more importantly, cosmic rays (neutrons) hitting silicon chips, creating charge on the nodes that flips a memory cell or a logic latch. These errors are transient and random. It is relatively easy to detect these errors in memories by protecting them with parity checking, and correcting these errors in memory is also relatively straightforward using error correcting code. However, if such a single-event upset occurs in a logic flip-flop, then it is difficult to detect and correct.

Researchers expect about an 8 percent increase in soft-error rate per logic state bit each technology generation.[7] Because the number of logic state bits on a chip double each technology generation (following Moore's law), the aggregate effect on soft-error rate failure in time of a chip is shown in Figure 5. Notice that by the 16-nm generation, the failure rate will be almost 100 times that at 180 nm.

Aging has had significant impact on transistor performance. Studies have shown that a transistor's saturation current degrades over years because of oxide wear out and hot-carrier degradation effects, as Figure 6 shows. So far, the degradation is small enough to account for as an upfront design margin in the VLSI component's specification. Researchers expect this degradation to become worse as we continue to scale transistor geometries beyond the 32-nm node. It might become so bad that it would be impractical to absorb degradation effects upfront in a system design.

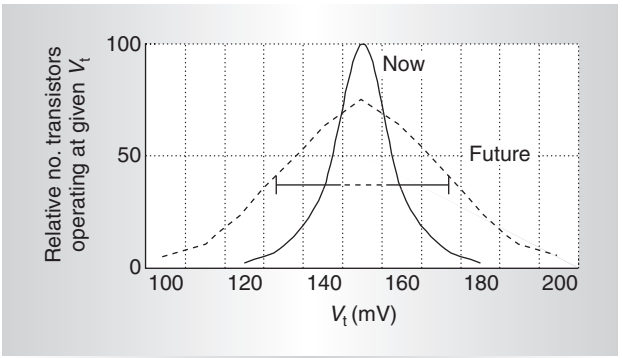

Figure 4. Extreme device variations will become more typical in the future.

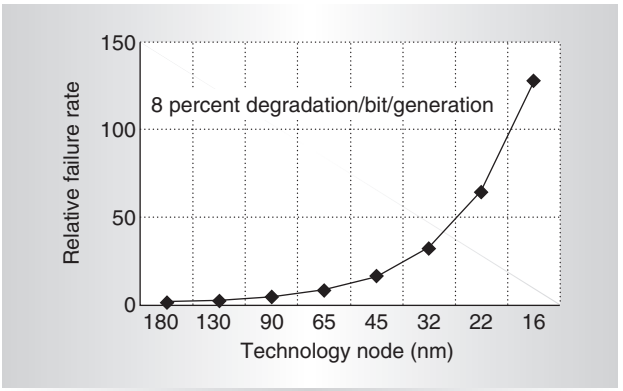

Figure 5. Soft-error failure-in-time of a chip (logic and memory).

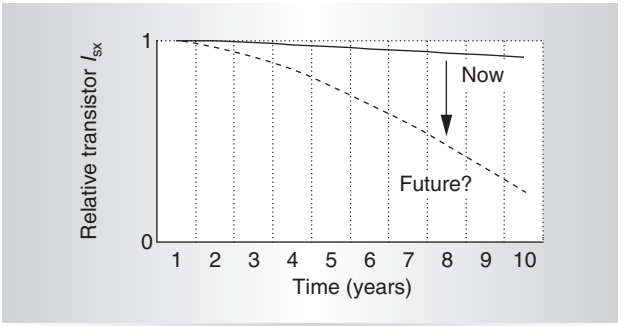

Figure 6. Time-dependent device degradation.

Impact on test

Future transistor scaling will have significant impact on test methodology as well. One important limiter is gate leakage. The gate dielectric (gate oxide) thickness must decrease as transistor dimensions scale downward to improve performance and reduce short-channel effects. However, as this oxide scales, density J_{ox} of the gate leakage current increases exponentially, as Figure 7 shows, because of quantum mechanical tunneling.

One preferred method of screening for defects and decreasing infant mortality in VLSI chips is called burn-in, where the chips are stressed with higher supply voltage at high-

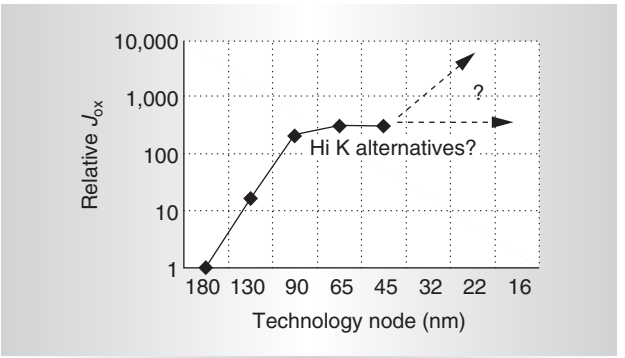

Figure 7. Density of gate leakage current increases exponentially in future technology generations, limiting the applicability of burn-in tests.

er temperature for a short period of time to accelerate aging. A fault in the chip shows up early during burn-in and hence is caught. Since the gate leakage current increases exponentially with supply voltage, leakage power during burn-in could become prohibitively high and could make burn-in testing obsolete.

A high dielectric constant (K) alternative is on the horizon to replace gate oxide. This high-K material should provide the same capacitance as silicon dioxide but with a much higher thickness; this thicker dielectric will have less leakage. However, this dielectric thickness, too, will scale downward over time and ultimately face the same gate leakage problem.

Because of extreme variability in transistor performance, the absence of burn-in testing to catch early defects, and latent defects caused by time-dependent device degradation, one-time factory testing of VLSI components after fabrication is insufficient—the VLSI component and the system could fail at any time in the field. Therefore, we need to devise new techniques to guarantee system reliability.

The reliability challenge

Putting it all together, a VLSI chip will have the following characteristics: It will have tens of billions of transistors, but many of them might be unusable because of extreme static variations. Furthermore, circuits will encounter dynamic variations of supply voltage and temperature; frequent and intermittent soft-errors; and transistors that slowly age and degrade over time, degrading circuit performance. Despite these difficulties and the fact that the chips cannot be retested at the factory, users expect the system to remain reliable and to continue to deliver the rated performance. This challenge will undoubtedly require a major paradigm shift in all aspects of VLSI design—fabrication, design, microarchitecture, testing, software, and applications.

Potential solutions

There are several potential solutions in sight in all disciplines to tackle most of the problems discussed here. However, all VLSI disciplines must work together toward successful solutions.

In VLSI design methodology, a shift from deterministic design to probabilistic and statistical design would ease the impact of transistor variations on circuit performance. Today's design optimizations can handle only one or two objectives, namely performance and power. This mindset will have to change, moving toward multivariable design optimizations that account for performance, active and leakage power, reliability, yield, and bin splits. We must work together to develop design tools to implement such optimizations, and statistical and probabilistic methodologies to go along with the tools.

In circuit design, replacing regular flip-flops by soft-error-tolerant hardened flip-flops will improve soft-error tolerance by almost 10 times. To catch dynamic errors, innovative techniques such as Razor[6] need serious consideration. Such techniques will not only detect and correct errors but will allow the design to operate at optimum power and performance. The Razor technique is power efficient because it does not replicate all the hardware, but only those flip-flops that are critical and require checking for correctness. This technique is also capable of catching circuit marginalities arising from transistor performance variations. In addition, algorithmic techniques at the functional-block level might be applicable to improve noise marginalities as well as cope with dynamic variations, thus improving reliability.[8]

At the system architecture level, the traditional method of error detection with functional redundancy checking might work; however, this method might not be power and energy efficient since it almost doubles the hardware and power consumption for the same performance. Designers must use any

redundancy and checking hardware judiciously to dynamically catch errors and take corrective action; such hardware should not burden the system with excessive power consumption and complexity. Called the Reliability and Security Engine, this is a comprehensive approach to an architectural framework.[9] In another interesting microarchitecture, a traditional processor core is accompanied by a small yet robust core as a checker.[10] The checker core is correct by construction, might be overdesigned to be variation tolerant, and is made immune from any further errors—both static and dynamic. Since the checker core is small, it consumes very little power and can dynamically detect and correct any errors made by the large core, thus providing reliable system operation.

The key to the reliability problem might be to exploit the abundance of transistors—use Moore's law to advantage. Instead of relying on higher and higher frequency of operation to deliver higher performance, a shift toward parallelism to deliver higher performance is in order, and thus *multi* might be the solution at all levels—from multiplicity of functional blocks in a design to multiple processor cores in a system.

Multiple functional blocks, operating at lower voltage and frequency, provide the same logic throughput, but at much reduced power, and can be used for redundancy and error checking. For example, a design could use two arithmetic and logic units to provide higher throughput when needed; each unit could check and correct results produced by the other. Multiple cores in a system will provide similar performance and redundancy benefits with functional redundancy checking employed at a coarse-grained level. For example, one core could check results produced by several cores; of course, software and applications will have to support this concept whenever possible.

We could distribute test functionality as a part of the hardware to dynamically detect errors, or to correct and isolate aging and faulty hardware. Or, a subset of cores in the multicore design could perform this task. This microarchitecture strategy, with multicores to assist in redundancy, is called *resilient microarchitecture*. It continually detects errors, isolates faults, confines faults, reconfigures the hardware, and thus adapts. If we can make such a strategy work, there is no need for one-time factory testing or burn-in, since the system is capable of testing and reconfiguring itself to make itself work reliably throughout its lifetime.

All this is possible because of the abundance of transistors, but all disciplines—from fabrication to software—will have to cooperate and make the system reliable in spite of unreliable components. A lot of research and development needs to happen, however, to make this concept a reality.

For reasons discussed before, the behavior of a VLSI chip and the system could look different from what designers intended: Static and dynamic variations will cause inherent unreliability in the design, traditional testing will be ineffective, and intermittent soft errors and long term device degradation will pose challenges to ensuring the overall system's reliability. The challenges painted here are for the extreme case: assuming the trends continue and that no innovations occur in VLSI system design. I am very optimistic that academic and industrial research will find innovative solutions for these challenges—as we always have in the past. That is why I intend this discussion to inspire research toward building reliable systems with unreliable components. MICRO

Acknowledgments

I thank Vivek De, Jim Tschanz, Ali Keshavarzi, Keith Bowman, Tanay Karnik, Peter Hazucha, Jose Maiz, and Shu-Ling Garver for their help and insightful discussions.

References

1. S. Borkar, "Design Challenges of Technology Scaling," *IEEE Micro,* vol. 19, no. 4, Jul.-Aug. 1999, pp. 23-29.
2. S. Borkar, "Circuit Techniques for Subthreshold Leakage Avoidance, Control, and Tolerance," *Proc. Int'l Electron Devices Meeting* (IEDM 2004), IEEE Press, pp. 421-424.
3. S. Borkar et al., "Parameter Variations and Impact on Circuits and Microarchitecture," *Proc. 40th Design Automation Conf.* (DAC 03), IEEE CS Press, 2003, pp. 338-342.
4. X. Tang, V. De, and J. Meindl, "Intrinsic MOSFET Parameter Fluctuations Due to

Random Dopant Placement," *IEEE Trans. Very Large Scale Integration (VLSI) Systems,* vol. 5, no. 4, Dec. 1997, pp. 369-376.
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. D. Ernst et al., "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," *Proc. 36th Int'l Symp. Microarchitecture* (Micro-36), IEEE CS Press, 2003, pp. 7-18.
7. P. Hazucha et al., "Neutron Soft Error Rate Measurements in a 90-nm CMOS Process and Scaling Trends in SRAM from 0.25-µs to 90-nm Generation," *Proc. Int'l Electron Devices Mtg.* (IEDM 2003), IEEE Press, pp. 21.5.1-21.5.4.
8. R. Hegde and N. Shanbhag, "Energy-Efficient Signal Processing via Algorithmic Noise-Tolerance," *Proc. Int'l Symp. Low-Power Electronics and Design* (ISPLED 99), IEEE Press, 1999, pp. 30-35.
9. 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.
10. T. Austin, "DIVA: A Reliable Substrate for Deep Submicron Microarchitecture Design," *Proc. 32nd Int'l Symp. Microarchitecture* (Micro-32), IEEE CS Press, 1999, pp. 196-207.

Shekhar Borkar is director of microprocessor research at Intel Corp. His research interests include high-performance and low-power digital circuits, and high-speed signaling. Borkar has an MSc in physics from the University of Bombay and an MSEE from the University of Notre Dame. He is a member of IEEE and an Intel Fellow.

Direct questions and comments about this article to Shekhar Borkar, 2111 N.E. 25th Ave., Hillsboro, OR 97124; shekhar.y.borkar@intel.com.