

# Modeling SRAM Failure Rates to Enable Fast, Dense, Low-Power Caches

Jangwoo Kim<sup>1</sup>

Mark McCartney<sup>2</sup>

Ken Mai<sup>2</sup>

Babak Falsafi<sup>3</sup>

<sup>1</sup>*Architecture Technology Group, Sun Microsystems*

<sup>2</sup>*Computer Architecture Laboratory (CALCM), Carnegie Mellon University*

<sup>3</sup>*Parallel Systems Architecture Lab (PARSA), École Polytechnique Fédérale de Lausanne*  
<http://www.ece.cmu.edu/~truss>

**Abstract**—The embedded memory hierarchy of microprocessors and systems-on-a-chip plays a critical role in the overall system performance, area, power, resilience, and yield. However, as process technologies scale down, the design of the embedded memory system is becoming increasingly difficult due to a number of exacerbating factors. Consequently, it is critical to provide accurate modeling tools for estimation of bit error rates in SRAM.

In this paper, we describe a modeling infrastructure which takes into account error-correcting codes (ECC) and redundancy to provide feedback on the resilience of a particular variability tolerance technique. We show that with the combination of inline multi-bit ECC for hard errors and a small amount of redundancy, performance, power consumption, stability, and yield can all be improved.

**Index Terms**—ECC, modeling, resilience, yield.

## I. INTRODUCTION

The embedded memory hierarchy of microprocessors and systems-on-a-chip plays a critical role in the overall system performance, area, and power. The cache hierarchy on a modern chip multiprocessor is often in the critical path of the design, can utilize over 50% of the die area, and can consume an appreciable fraction of the total power. Thus, the resilience and yield of the memory system is crucial to the resilience and yield of the entire system. However, as process technologies scale down to nanometer-regime geometries, the design and implementation of the embedded memory system is becoming increasingly difficult due to a number of factors including increasing process variability, manufacturing defects, device wearout, and susceptibility to energetic particle strikes. Of particular concern are variability-induced hard errors, due to the high susceptibility of static random access memory (SRAM) to variability effects due to the large number of near minimum-sized devices in the cell arrays [1].

In this paper, we explore the potential of using ECC and a small amount of redundancy to enable dense, low-power caches through detailed SRAM

failure modeling and statistical analysis. We also present the ideal coding strength and redundancy requirements for optimal failure coverage. In Section II, we detail our transistor-level SRAM variation simulation methodology. In Section III, we explore the results and demonstrate that if single-bit error tolerance is combined with 3% redundancy, the memory can be set for ~35% faster read access, ~80% faster write access, ~85% higher read upset margin, and ~24% lower minimum voltage, while maintaining yield.

## II. SRAM FAILURE RATE METHODOLOGY

Process scaling leads to numerous sources of intra-die variation, which researchers categorize as either spatially correlated, systematic variation or random mismatch [2]. The resulting circuit-level effects of both spatially correlated and random mismatch variations can be represented as shifts in the transistor's threshold voltage ( $V_{th}$ ) and drive current. In this section, we first review the failure modes of SRAM cells due to intra-die variability. Then, we discuss our methodology for simulation to determine the probability of these failures.

*Prior work in variability error modeling:* To assess the yield of future cache designs, researchers have modeled device variability in current and future technologies, and reported estimated yields using Monte Carlo simulations of both analytic and circuit models [3]-[5]. However, no known prior work investigated the potentials of using multi-bit ECC in combination with redundancy in detail to improve a combination of performance, power consumption, stability, and yield, as done in this paper.

*SRAM cell failure modes:* Variability can cause four failure modes in SRAM cells: read stability failure, read latency failure, write latency failure, and minimum hold voltage failure [6]. A read stability failure occurs during a read access, when current flows from the precharged bitline through access and pull-down transistors, and the nominally low voltage at one

storage node bumps up sufficiently to exceed the trip point of the inverter, inadvertently flipping the cell's stored value. A read latency failure occurs during a read access, when the cell fails to pull down one of the bitlines by a large enough voltage to overcome the sense amplifier offset and other noise sources within a target latency, meaning the value stored in the cell cannot be correctly read at the target performance point. A write latency failure occurs during a write access, when the high voltage storage node cannot be pulled below the trip point of the inverter to flip the stored value in the target latency, meaning the cell won't flip and the write fails. A minimum hold voltage failure occurs during the time when an SRAM cell is not accessed, if the supply voltage is reduced below a minimum level which corrupts the cell's data.

*SRAM cell model:* Considering the four failure cases above, we developed an SRAM cell analysis model derived from the work in [5]-[7]. We replaced the square-law current models with alpha-law current models, using an alpha of 1.3, to capture velocity saturation and mobility degradation in short channel transistors [8]. We assume a conventional symmetric 6T SRAM cell with differential bitlines for our modeling. Our analytical model characterizes cell performance by measuring read stability, read latency, write latency, and minimum hold voltage. To determine the cell's read stability, we measure the voltage difference between the voltage rise at the read node and the trip point of the other side. We measure the cell's read latency as the time taken for the precharged bitlines to separate by a certain amount. We measure the write latency as the time taken for the voltage at the right node to drop to a point that upsets the cell's data. We

measure the minimum hold voltage as the minimum supply voltage below which the stored value is corrupted. Because we assume a symmetric cell design with differential read and write, we measure each metric on both sides of the cell and take the larger failure rate of the two sides to be the failure probability of that cell. The results from our analytical cell failure model closely match circuit simulation and the results reported in the literature [6].

*SRAM array model:* We implement a 4-way set-associative 32KB cache, which consists of 256x256 subarrays (derived from CACTI [9]). While we chose a 32KB SRAM for our analysis, our results can be extrapolated to larger caches, since modern microprocessors often contain large caches combining many small subarrays to reduce the access latency and leakage power [10]. We sample each cell using Monte Carlo simulations of our analytical cell model with variations in device parameters. When an array is constructed, we consider the array's read stability, read and write latency, and minimum hold voltage as those of the worst-case cell within the array.

*Variations in device parameters:* Table 1 lists the mean and standard deviation sizes and threshold voltages of three types of transistors used in a standard 6T SRAM cell in three different technology nodes. The baseline DRC-compliant cell transistor sizes were chosen by circuit simulation in an industrial 45nm process; sizes for the other technology nodes were extrapolated assuming a critical dimension scaling trend of roughly 0.7. We calculated the effective gate length from the drawn gate length by subtracting a fixed length (the poly undercut) which is alluded to in ITRS projections [12] as the difference between drawn

**Table 1. Nominal values and variations of baseline device parameters obtained using an industrial 45nm technology (Vdd = 1V, Alpha-power model ( $\alpha=1.3$ ), Low power: Vdd=0.7V, Low area: 20% smaller cell)**

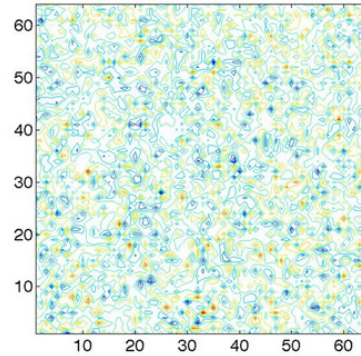
|                          |            | Pull-Down Transistor |  |       | Access Transistor |  |       | Pull-Up Transistor |  |       |
|--------------------------|------------|----------------------|--|-------|-------------------|--|-------|--------------------|--|-------|
|                          |            | W                    | $L_{\text{drawn}}/ L_{\text{effective}}$ | Vth   | W                 | $L_{\text{drawn}}/ L_{\text{effective}}$ | Vth   | W                  | $L_{\text{drawn}}/ L_{\text{effective}}$ | Vth   |
| 45nm baseline, low power | Mean (nm)  | 200                  | 60/43                                    | 290   | 120               | 60/43                                    | 290   | 120                | 60/43                                    | 290   |
|                          | 3 $\sigma$ | 1.3%                 | 6.05%                                    | 33.5% | 2.17%             | 6.05%                                    | 43.2% | 2.17%              | 6.05%                                    | 43.2% |
| 45nm low area            | Mean (nm)  | 140                  | 45/28                                    | 290   | 85                | 45/28                                    | 290   | 85                 | 45/28                                    | 290   |
|                          | 3 $\sigma$ | 1.86%                | 9.29%                                    | 49.6% | 3.06%             | 9.29%                                    | 63.6% | 3.06%              | 9.29%                                    | 63.6% |
| 32nm [11]                | Mean (nm)  | 140                  | 45/33                                    | 290   | 85                | 45/33                                    | 290   | 85                 | 45/33                                    | 290   |
|                          | 3 $\sigma$ | 1.36%                | 5.76%                                    | 45.7% | 2.24%             | 5.76%                                    | 58.6% | 2.24%              | 5.76%                                    | 58.6% |
| 22nm [11]                | Mean (nm)  | 100                  | 30/22                                    | 290   | 60                | 30/22                                    | 290   | 60                 | 30/22                                    | 290   |
|                          | 3 $\sigma$ | 1.30%                | 5.91%                                    | 66.2% | 2.17%             | 5.91%                                    | 85.4% | 2.17%              | 5.91%                                    | 85.4% |

and physical gate length. Similarly, we extracted variability in critical dimensions using the gate CD control figure from ITRS [12]. Given the gate width and effective length, we applied the Pelgrom model [13] to calculate variations in threshold voltage using an Avt of  $3\text{mV}\cdot\mu\text{m}$  [14].

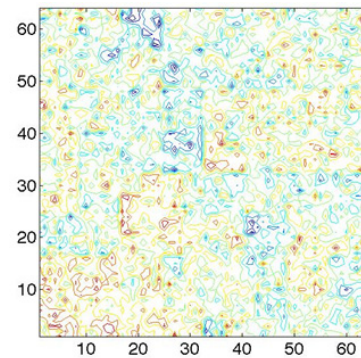
*Variation types:* Intra-die variations are a combination of random device mismatch and spatially correlated variations [15]. Since random mismatch is becoming the most dominant source of intra-die variation in future technologies [12], we separately modeled two intra-die variation scenarios: intra-die variation caused by only random mismatch, and a combination of both random mismatch and spatially correlated variation, assuming the overall variation is equally divided between the two. Figure 1(a) shows intra-die variation caused by only random mismatch, while Figure 1(b) combines both random mismatch and spatially correlated variation. We applied spatially correlated variation using a multi-level hierarchy of independent random variables as shown in Figure 1(c) [16]. For example, the spatially correlated variation in a 32KB array is modeled by combining  $\sim 350\text{K}$  random variables of device variation parameters from 9 levels of hierarchical arrays.

### III. RESULTS AND ANALYSIS

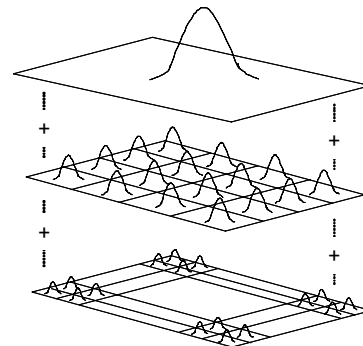
Memory system designers have considered using error-correcting codes (ECC), already deployed for soft error protection, to correct hard errors as an effective low-overhead variability mitigation technique. If a low overhead multi-bit ECC technique were possible and used to correct hard manufacture-time errors, the cache performance, power, and area could be significantly improved. Figure 2 shows distributions of SRAM arrays for four parameters under intra-die variations of both random mismatch and spatially correlated variation modeled for 45nm process technology. Each SRAM array is equipped with conventional (72,64) SECDED ECC and 3% redundant rows. The 'baseline' design does not employ ECC or row redundancy to handle variability, and thus the overall yield is determined by that variability tolerance of the worst-case cell within an SRAM. The "SEC only" design uses the existing SECDED ECC to mask out the worst-case cells per word for each failure category. Therefore, the worst of the second-worst cells per word determines the chip failure point. The "SEC + 3%" design uses both ECC and row redundancy to mask out up to 3% of the second-worst cells per word in addition to the worst cells per word. The X-axis is normalized to



(a) Variations due to random mismatch



(b) Variations due to both random mismatch and spatially correlated variation

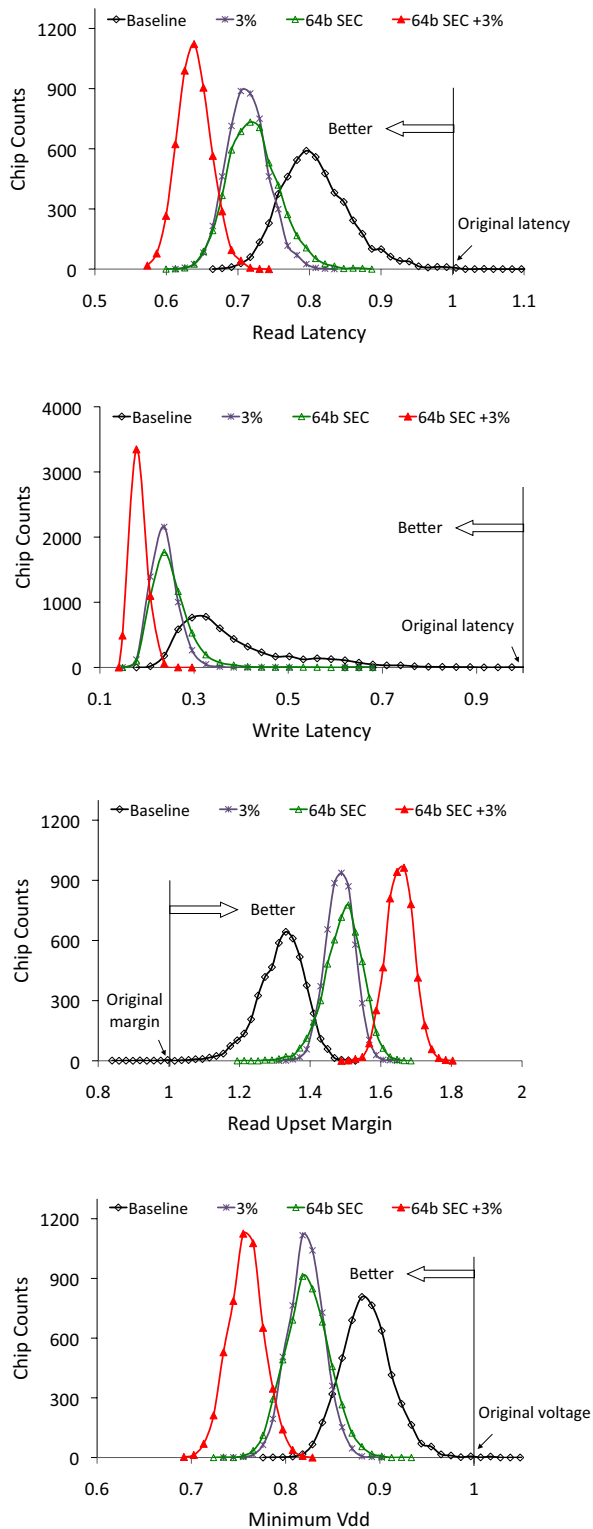


(c) Hierarchical model for spatially correlated variation

**Figure 1. Intra-die variation.** The figure (a) shows the intra-die variation due to only random mismatch. (b) shows the intra-die variation due to both random mismatch and spatially correlated variation equally splitted. (c) shows spatially correlated variation modeled by a multi-level hierarchy of independent variables. The overall variations are the same for both models. The grids shown here are  $64 \times 64$ .

the  $3\sigma$  point of each baseline distribution and the Y-axis shows the chip count at the failure point.

The results highlight key advantages of using ECC to correct variability errors. First, SECDED ECC is



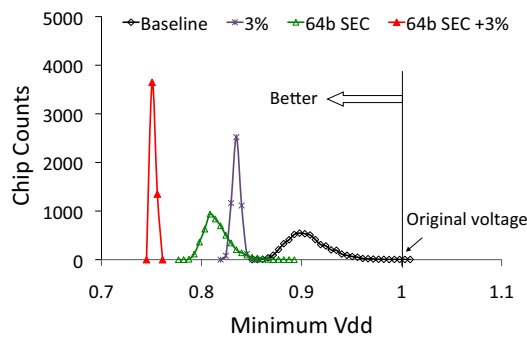
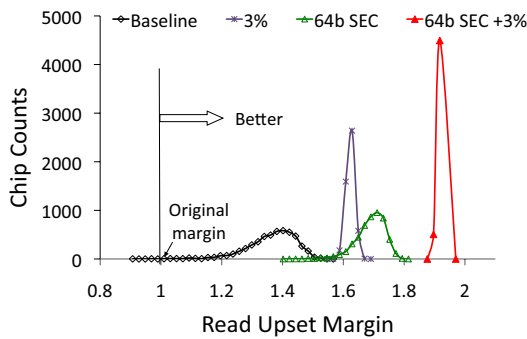
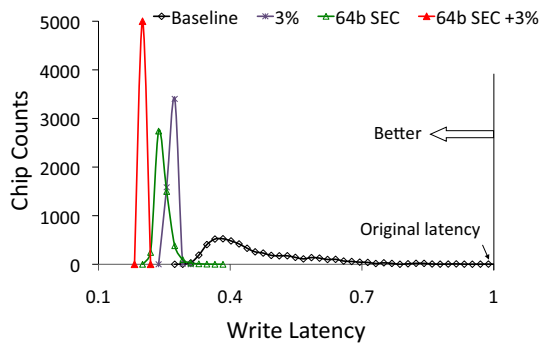
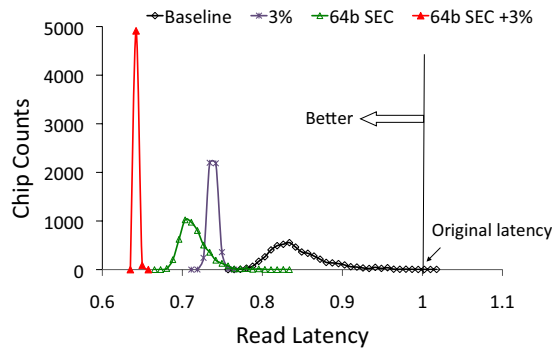
**Figure 2. Intra-die variations using both random mismatch and spatially correlated variation.** Distributions of 5,000 32KB SRAMs: (a) read latency (b) write latency, (c) read stability, and (d) minimum hold voltage

very effective in reducing yield loss due to variability because the baseline distribution is shifted in the direction of faster read and write performance, higher read stability, and lower voltage. The variance within a new distribution is also narrower. Second, when ECC is combined with a small amount of line redundancy, the baseline distribution is shifted much further. This degree of shift indicates that yield loss due to variability in the baseline design is completely eliminated so that memory system designers can improve performance, stability, and voltage of the same SRAM without experiencing any yield loss. Third, when we use ECC for variability tolerance, we only need a small amount of line redundancy because the redundancy is only used for correcting a small number of words with multiple variability errors.

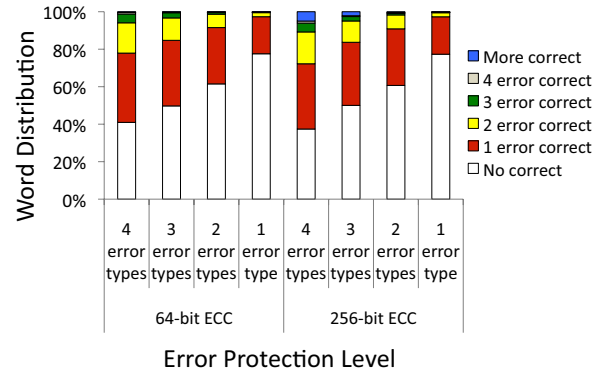
Figure 3 shows the distributions obtained when intra-die variability is modeled by only random mismatch. Because random mismatch is expected to become the most dominant source of variability in future technologies, we can see the potential of using ECC to tolerate variability errors in the future. These results actually indicate stronger advantages for ECC under heavy random mismatch variation. When ECC is combined with 3% redundancy, the memory can be set for ~35% faster read access, ~80% faster write access, ~85% higher read upset margin, and ~24% lower minimum voltage, while maintaining equal or better yield to the baseline. ECC's strong variability tolerance under heavy mismatch comes from its effectiveness in handling randomly distributed small-scale errors. Spatially correlated variation leads to local clustering of variations within memory where multiple cells within a same word are likely to share a similar level of variability. This clustering limits the effectiveness of ECC.

Table 2 shows the variability tolerance based on the amount of redundancy given to SECDED ECC for each failure mode. We observe a combination of SECDED with 3% of redundancy as an optimal point.

Figure 4 shows an average distribution of all words in a single SRAM based on the number of bad cells that the ECC must correct to achieve target variability tolerance in '3%' column in Table 2 and 'SEC+3%' in Figure 3. The X-axis shows the various target error coverages. This result shows the actual amount of redundancy required to achieve the target coverage and we observe that it is not necessary to dedicate SECDED ECC and 3% of redundancy to each failure mode because errors are randomly distributed across all words within the SRAM array and there are not many words containing more than 2 bad cells.



**Figure 3. Intra-die variations using only random mismatch.** Distributions of 5,000 32KB SRAMs: (a) read latency (b) write latency, (c) read stability, and (d) minimum hold voltage

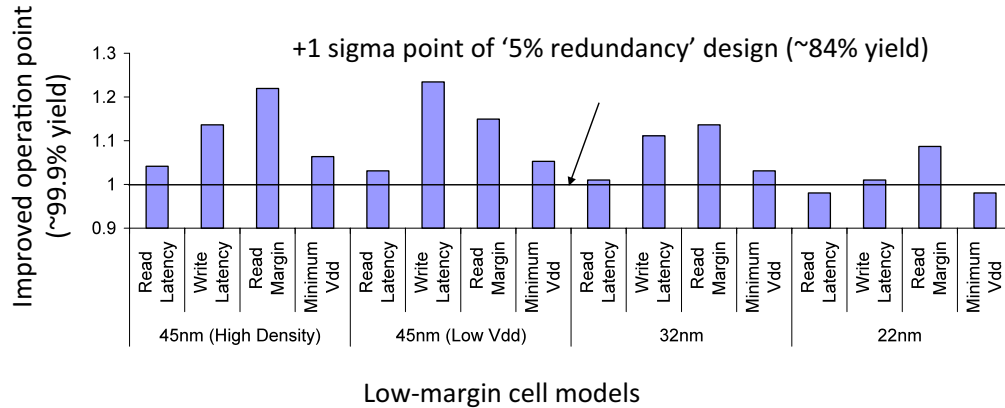


**Figure 4. Error correction distribution.** Word distribution based on the number of bad cells obtained from simulation results for Figure 3.

*Cell scaling results:* Figure 5 shows that if we can correct the worst cell from each word, the cache performance, read margin, power, and yield improve. Each bar represents a new operation point of the baseline design at  $+3\sigma$  using ECC for variability tolerance. The X-axis lists four failure cases of each cell model. The Y-axis is normalized to operation points at  $+1\sigma$  of the baseline design. Therefore, if the new operation point is higher than 1, it means that the new design performs better for that category and improves the yield cut from  $+1\sigma$  ( $\sim 84\%$  yield) to  $+3\sigma$  ( $\sim 99.9\%$  yield). The results show that the use of low-overhead inline error correction enables caches using an aggressively sized or low-power cell or scaling down to reasonable predictions of 32nm technology [11], while improving both yield and operating points of the baseline design.

**Table 2. Variability tolerance improvement per failure mode based on the amount of redundancy allocated**

| Word Size | Failure Mode  | Redundancy |            |     |     |
|-----------|---------------|------------|------------|-----|-----|
|           |               | None       | 3%         | 5%  | 10% |
| 64b       | Read Margin   | 42%        | <b>85%</b> | 89% | 92% |
|           | Read Latency  | 20%        | <b>36%</b> | 37% | 38% |
|           | Write Latency | 68%        | <b>80%</b> | 80% | 81% |
|           | Min. Voltage  | 11%        | <b>24%</b> | 25% | 26% |
| 256b      | Read Margin   | 40%        | <b>75%</b> | 79% | 83% |
|           | Read Latency  | 18%        | <b>31%</b> | 32% | 34% |
|           | Write Latency | 64%        | <b>79%</b> | 79% | 80% |
|           | Min. Voltage  | 10%        | <b>20%</b> | 21% | 22% |



**Figure 5. Improving yield, performance, stability, and power of aggressively designed cells with low margins.** Y-axis is normalized to the operation point at  $+1\sigma$  of ‘baseline of +5%’ redundancy design. Each bar shows the operation point at  $+3\sigma$  of baseline design using DECTED ECC to correct variability errors.

#### IV. CONCLUSIONS

In this work, we introduced a modeling infrastructure to calculate SRAM cell failure rates and determine the potential of ECC to combat variation-induced memory cell failures. We showed how single-bit hard error tolerance can be traded off for higher-density bitcells, higher performance, greater cell stability, or lower power design, while maintaining high yield.

#### REFERENCES

- [1] R. Heald and P. Wang, “Variability in sub-100nm SRAM designs,” in *IEEE/ACM International Conference on Computer Aided Design*, 7-11 Nov. 2004, pp. 347-352.
- [2] K. Bowman, S. Duvall, and J. Meindl, “Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration,” *IEEE J. Solid-State Circuits*, vol. 37, no. 2, pp. 183-190, 2002.
- [3] A. Agarwal, B. Paul, S. Mukhopadhyay, and K. Roy, “Process variation in embedded memories: failure analysis and variation aware architecture,” *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1804-1814, Sep. 2005.
- [4] S. Ozdemir, D. Sinha, G. Memik, J. Adams, and H. Zhou, “Yield-aware cache architectures,” in *IEEE/ACM International Symposium on Microarchitecture*, Dec. 2006, pp. 15-25.
- [5] S. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, “VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects,” *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 1, pp. 3-13, Feb. 2008.
- [6] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, “Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, pp. 1859-1880, Dec. 2005.
- [7] A. Chandrakasan, W. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. IEEE Press, 2000.
- [8] T. Sakurai and A. Newton, “Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas,” *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584-594, 1990.
- [9] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, “Cacti 5.1,” HP Laboratories, Palo Alto, Tech. Rep. 20, Apr. 2008.
- [10] J. Chang, M. Huang, J. Shoemaker, J. Benoit, S.-L. Chen, W. Chen, S. Chiu, R. Ganesan, G. Leong, V. Lukka, S. Rusu, and D. Srivastava, “The 65-nm 16-MB Shared On-Die L3 Cache for the Dual-Core Intel Xeon Processor 7100 Series,” *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 846-852, 2007.
- [11] W. Zhao and Y. Cao, “New generation of predictive technology model for sub-45nm design exploration,” in *Proc. International Symposium on Quality Electronic Design*, 27-29 Mar. 2006.
- [12] “International Technology Roadmap for Semiconductors,” Semiconductor Research Corporation, Tech. Rep., 2007.
- [13] M. Pelgrom, A. Duinmaijer, and A. Welbers, “Matching properties of mos transistors,” *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433-1439, Oct. 1989.
- [14] J. Croon, S. Decoutere, W. Sansen, and H. Maes, “Physical modeling and prediction of the matching properties of mosfets,” in *Proc. European Solid-State Device Research Conference*, 21-23 Sep. 2004, pp. 193-196.
- [15] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, “Parameter variations and impact on circuits and microarchitecture,” in *Proc. Design Automation Conference*, 2003, pp. 338-342.
- [16] A. Agarwal, D. Blaauw, and V. Zolotov, “Statistical timing analysis for intra-die process variations with spatial correlations,” in *Proc. International Conference on Computer Aided Design*, 2003, pp. 900-907.