Selective Wordline Voltage Boosting for Caches to Manage Yield under Process Variations

Yan Pan†, Joonho Kong‡, Serkan Ozdemir†, Gokhan Memik†, Sung Woo Chung‡
†Northwestern University
2145 Sheridan Road, Evanston, IL
{panyan,s-ozdemir,g-memik}@northwestern.edu
‡Korea University
Anam-dong Seongbuk-Gu, Seoul, 136-713 Korea
{luisfigo77,swchung}@korea.ac.kr

ABSTRACT
One of the most important hurdles of technology scaling is process variations, i.e., variations in device characteristics. Process variations cause large fluctuations in performance and power consumption in the manufactured chips. In addition, these fluctuations cause reductions in the chip yields. In this work, we present an analysis of a representative high-performance processor architecture and show that the caches have the highest probability of causing yield losses under process variations. We then propose a novel selective wordline voltage boosting mechanism that aims at reducing the latency of the cache lines that are affected by process variations. We show that our approach can eliminate over 80% of the yield losses under medium level of variations, while incurring less than 1% per-access energy overhead on average and less than 4.5% area overhead.

Categories and Subject Descriptors
B.3.2 [**Memory Structures**]: Design Styles - *cache memories*

General Terms
Performance, Design, Reliability

Keywords
Cache, Process Variations, Yield, Access Time Failure, Selective Wordline Voltage Boosting

1. INTRODUCTION
One of the major challenges faced by deep submicron technologies is process variations, which adversely affect performance and power consumption, and hurt yield [30]. Among the various factors that affect yield, parametric yield losses are the most dominant factor; with smaller technologies, parametric yield losses constitute more than 50% of the losses [21].

Although various design components are affected by process variations, caches are the most vulnerable. The 6T SRAM cell represents the finest feature sizes for a fabrication technology; in fact, foundries usually use 6T SRAM arrays for technology qualification. In addition, according to FMAX theory [4], components with high number of parallel and shallow critical paths, such as caches, are most susceptible to process variations. Caches also occupy a large share of chip area; hence they face larger variations. Any single failing SRAM cell can directly affect yield if no preventive techniques are adopted.

There are three kinds of SRAM cell failures: unstable read, unstable write, and access time failures. As we will further discuss in Section 2, among these failures types, the access time failures are the most crucial [1]. Access time failure means the actual access time to the cache exceeds the pre-defined access time. As variation in threshold voltages amplifies, the probability of the access time failures increases [1]. In case of higher level caches (level 2 and 3), NUCA cache architecture [13] can be adopted to endure different cache access times due to process variations and hence eliminate the effect of increased access times on yields. Another conventional way to mitigate process variations in caches is to employ redundancy schemes [5][8][11][19][25][29][32]. However, such schemes have considerable overhead. First, the redundant lines cause significant area overhead. In addition, the number of redundant cache lines may have to be increased significantly to cover all faulty lines under severe process variations, particularly considering that the redundant lines themselves are prone to process variations.

In this work, we propose a simple, yet efficient, technique to mitigate process variations by selective voltage boosting. By selectively boosting the voltage of components on the failing critical paths, the cache access times can be significantly reduced, while still maintaining small power and area overheads. The proposed technique can be applied to any SRAM array structure, however, in this work, we focus on level 1 (L1) caches since their access times have a direct impact on the chip yield.

The rest of this paper is organized as follows. In Section 2, we present our analysis showing that caches are indeed the most susceptible components to process variations. We demonstrate our proposed technique in Section 3, while the evaluation results in terms of yield, power, and area costs are provided in Section 4. A review of related works is presented in Section 5 and we conclude our paper in Section 6.

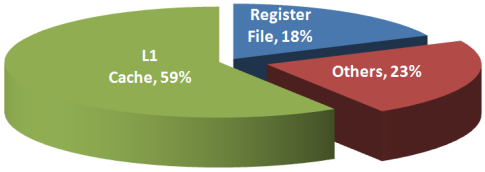


Figure 1. Distribution of critical paths in a representative processor architecture.

Table 1. Probability of cache failures w.r.t. σV_{th} [2]. P(x) indicates the probability of failure 'x' occurring.

σV_{th}	P(read failure)	P(write failure)	P(access time failure)	P(cache failure)
20	1E-4	1E-4	*1E-4*	1E-4
30	2E-4	2E-4	*5E-4*	8E-4
40	1E-3	6E-4	*3E-3*	4E-3
50	5.8E-3	3.8E-3	*1.4E-2*	2.2E-2

DAC'09, July 26-31, 2009, San Francisco, California, USA

2. ANALYSIS OF YIELD LOSSES

To analyze the impact of process variations on a processor architecture, we carried out a Monte Carlo study on 1000 chips modeled after an Alpha 21364 (EV7) processor using the framework described in Section 4.1. We considered the critical paths in each pipeline stage and run simulations for the branch predictor, register rename unit, issue queue, register file, integer execution unit, and L1 data cache [15]. Figure 1 shows the distribution of the critical paths among various pipeline components. Our results reveal that 59% of the critical path lies in the L1 caches. This is not a surprising result, as previous work [4] has shown that structures with many independent paths with low logic depth are most likely affected by parametric variations. Thus, we focus on implementing our voltage boosting scheme only on caches. Although it is not depicted here, L2/L3 caches are also highly affected by process variations. However, since they are usually not on the processor critical path, other techniques such as NUCA can be adopted to alleviate the increase in access latencies in L2/L3 caches.

SRAM cells have been used as primary storage in microprocessors. Currently, they are widely used in caches, register files, etc. due to many advantages of SRAM cells (such as fast access speed, no need to refresh, and stability). However, as process variations become severe, SRAM cells become vulnerable in terms of their stability and access speed. Table 1 depicts the relation between the probabilities of cache failures and the standard deviation of V_{th} (σV_{th}) [2]. When σV_{th} is 20mV, the probability of three kinds of cache failure is 1E-4. However, as the σV_{th} is increased, the access time failure becomes dominant. Furthermore, the situation gets worse in the cache line granularity. Liang et al. [17] simulated line-level failure rate for a cache with 32 Bytes linesize and observed a 64% cache line failure in 32nm technology. Although redundant cache lines can replace these faulty cache lines, 64% cache line failure will be hard to recover from. This calls for a multitude of techniques to mitigate cache access time failures under process variations. These results also suggest that we can expect a significant yield improvement when we alleviate the cache access time failure.

3. SELECTIVE VOLTAGE BOOSTING

3.1 Design philosophy

3.1.1 Delay Reduction by Boosted Voltage

In a synchronized design, the longest pipeline stage determines the frequency of the processor. If this frequency is below the pre-determined minimum processor frequency, the processor becomes parametric yield loss, despite the fact that it might be functional. The main goal in this work is to reduce the cache access latency if it causes a parametric yield loss. Specifically, we propose to save the failing paths by boosting their power supply voltage.

The relation between power supply voltage and delay is described by Alpha Law [26]:

$$Delay \propto \frac{V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \quad (1)$$

Here 'α' is a technology dependent constant that is greater than 1. Hence, without tuning the V_{th} in the circuit, the delay can be reduced by boosting the V_{dd}.

3.1.2 Selective Wordline Voltage Boosting

The reduced delay by boosting supply voltage does not come for free. It incurs increased dynamic and static power. Hence, careful design compromises must be made to achieve an efficient implementation. In our design, we propose to *only boost the voltage on selected wordlines*. There are several reasons for this design decision. First, the delay on the long wordlines is a major component of the overall cache access time. Boosted supply voltage in its drivers can directly reduce this delay. Second, boosted wordline voltage helps to enhance read current that goes through the pass gates (PG) of the cells, hence helps to discharge the bitlines faster. Third, the leakage power is low on wordlines as they only affect gate leakage to the PG's. Thus, boosted voltage will not significantly increase the overall leakage power of the cache. Finally, failing cells are naturally grouped into rows that share the same wordline. Therefore, selective wordline boosting effectively implements boosting of only the failing rows.

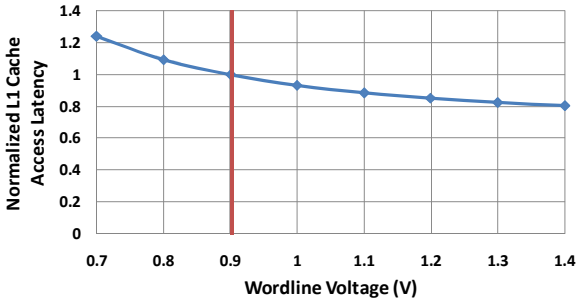


Figure 2. The relation between wordline voltage and cache delay. The results are obtained using SPICE simulations.

To estimate the efficiency of boosting only the wordline voltage, we simulated the access latency versus various wordline voltages using the framework described in Section 4.1. The results are shown in Figure 2. As the wordline voltage is increased from 0.9V to 1.3V, the L1 cache access latency is reduced by 18%. This result shows that the wordline voltage has the potential of being effectively used to reduce the access latency of caches.

3.2 Overall architecture

With the aim of boosting the voltage on selected wordlines, we propose the architecture shown in Figure 3. Only the circuits associated with a single wordline is illustrated as the remaining circuit design is not affected by our optimization. During chip testing, the rows in the L1 cache that fail the timing test are marked in a special purpose register file, where a single fault bit is dedicated for each wordline. In addition to the dedicated register, by using off-chip EEPROM, this failure information can be kept across system power down. The fault bits are used to drive the power selection circuit for the wordline buffers. Failing wordlines

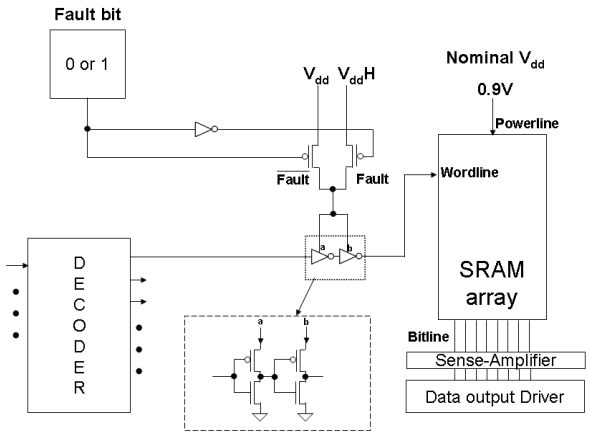


Figure 3. Proposed wordline voltage boosting architecture.

will adopt higher power supply voltage ($V_{dd}H$) in its drivers (the inverters), while passing wordlines will be fed with the nominal voltage (V_{dd}). The remaining components in the L1 cache all work under nominal power supply. In our evaluation, we target a 32KB L1 cache with wordline/bitline slicing, where 1024 separate wordlines are present. Thus 128 Bytes of fault bit storage (1024 bits), together with 2048 power gating PMOS's are added to the cache. Utilizing fault bits in our cache architecture makes sure that only wordlines of the faulty rows are raised high, hence the dynamic and static power overhead of our approach remains low. The distribution of $V_{dd}H$ could increase power routing complexity. However, this can be alleviated if techniques in [14] is used.

As the decoders are still working at the nominal voltage, while the wordline buffers can be raised to $V_{dd}H$, there is a voltage mismatch. However, there is no need to employ a level shifter. To achieve full rail (0~$V_{dd}H$) signal on the wordline, the supply voltage of both inverters are boosted to $V_{dd}H$. There is still increased leakage going through the PMOS of the first inverter due to incomplete shut-off of the PMOS (i.e., when input is V_{dd}, supply is $V_{dd}H$). We fully modeled this leakage and evaluated it to be minimal as compared to the total leakage in the L1 cache which is dominated by the SRAM array leakage (a detailed leakage overhead is presented in Section 4.1). Hence we chose not to adopt level shifters to avoid extra area overhead and additional latency on the critical path.

4. EVALUATION

4.1 Evaluation Methodology

To evaluate the efficiency of our proposed technique, we built a SPICE simulation framework that models process variations. In addition, a detailed architectural simulation framework (described in Section 4.1.3) is adopted to model the effect of our approach on the processor.

4.1.1 Circuit Model for L1 Cache

We adopt the floorplan and circuit structure of a 32KB L1 cache as described in CACTI 5.3 [35]. One critical path for the data array is shown in Figure 4.

A path consists of address H-Tree, address pre-decoder, address post-decoder, wordline driver, wordline, SRAM cell, bitline, precharge circuit, sense amplifier, multiplexer, and output data H-Tree. H-Trees, wordlines, and bitlines are modeled as RC lines, with repeaters inserted in the H-Trees. The nominal parameter values for the MOSFETs and interconnects are extracted from 45nm PTM [23]. The dimensions of the SRAM cells are based on the design by Hamzaoglu et al. [11]. The SPICE model without process variations is calibrated using CACTI 5.3.

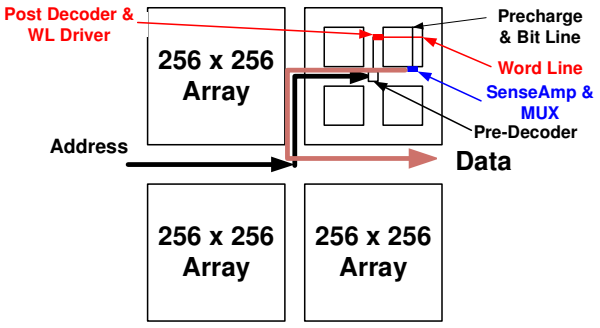


Figure 4. L1 Cache data array components: components on one path are shown.

Table 2. Nominal and 3σ variation values for each source of process variations modeled.

	Gate Length	Threshold Voltage	Metal Width	Metal Thickness	ILD Thickness
Nominal Value	45 nm	220 mV	0.25 µm	0.55 µm	2.5 nm
Low Var. [%]	±6.6	±12	±22	±22	±23
Medium Var. [%]	±10	±18	±33	±33	±35
High Var. [%]	±13.3	±24	±44	±44	±46

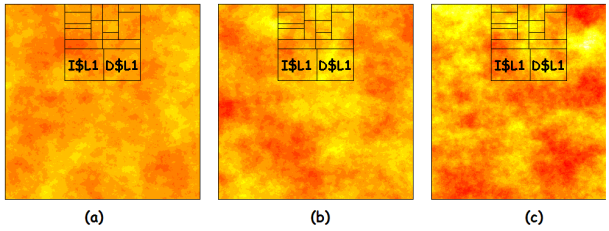


Figure 5. V_{th} variation maps for (a) low, (b) medium, and (c) high process variations.

4.1.2 Process Variation Modeling

Processes like sub-wavelength lithography and aggressive technology scaling result in statistical variations in circuit parameters such as gate-oxide thickness, channel length, and Random Doping Effects (RDE) [3]. These parametric variations can be classified into die-to-die (D2D) variations and within-die (WID) variations. D2D variation refers to the variation in process parameters across dies and wafers, whereas WID variation takes place in device features within a single die. Parametric variations can be of two categories: spatially-correlated (systematic) variations where devices close to each other have a higher probability of observing a similar variation level, and random (uncorrelated) variations causing random differences between various devices within a die. In this work, we model both systematic and random parametric variations.

To effectively model parametric variations, we account for five different variation parameters: metal thickness (T), inter-layer dielectric thickness (ILD or H), line-width (W) on interconnects, gate length (L_{gate}) and threshold voltage (V_{th}) for the MOS devices. We use the variation limits given by Nassif [20] as shown in Table 2 as our base case (Medium Variation). We also study two other cases where process variations are less and more severe (Low and High Variation, respectively); the parameter variations for those cases are also given in Table 2. Three sample maps corresponding to the V_{th} in the three process variation levels are presented in Figure 5. Note that high process variation corresponds to a larger span of colors on the map, and vice versa. To take into account the spatial correlation, we use a range factor (φ) in the two dimensional layout of the chip. Thus, each process parameter can be expressed as a function of its mean (µ), standard deviation (σ), and the range (φ) values. In this work, we used a

Table 3. Normalized dynamic energy and leakage power consumption of our proposed logic.

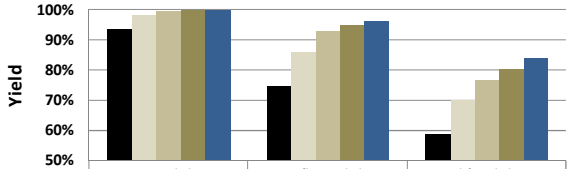
Scheme	Dynamic energy	Leakage power
Baseline 0.9V	1.0000	1.0000
Selective 1.0V	1.0216	1.0064
Selective 1.1V	1.0419	1.0076
Selective 1.2V	1.0685	1.0092
Selective 1.3V	1.0977	1.0125
Whole 1.0V	1.3649	1.6172

range factor of 0.2. With this background, we have generated a spatial map of parameter values using the VARIUS process variation model [27] implemented in the R statistical tool [34]. We divide the circuit floorplan into a grid of 5000 x 5000 points. Our framework then picks the process variation values on this grid and maps them to the RC line sections and MOSFETS in the SPICE model for simulation.

We use the Monte Carlo method to model a batch of chips and study the impact of process variations on cache yield. We simulated a total of 1000 chips for each process variation severity level. The cache access latency under process variations is simulated across all the paths, together with dynamic and leakage power statistics.

4.1.3 Architectural Simulation

To evaluate the energy consumption impact, we extracted cache access traces of SPEC2000 benchmark applications using SimpleScalar 3.0 Alpha simulator [31]. Using per access energy consumption and leakage power from the SPICE simulations, we calculated overall cache energy consumption including both leakage and dynamic energy for L1 data and instruction caches. In the following section, we present results for eight representative applications (applu, crafty, fma3d, gcc, gzip, mcf, mesa, and twolf). SimPoint [28] is used to improve the accuracy of the simulations. Data and instruction caches are 32KB in size, 2-way set associative, and have 32-byte linesizes. The clock frequency of the simulated processor is set to be 3.5 GHz.



	Low Variation	Medium Variation	High Variation
Baseline	93.7%	74.7%	58.8%
VddWL=1.0V	98.0%	85.9%	70.0%
VddWL=1.1V	99.4%	93.1%	76.7%
VddWL=1.2V	99.8%	94.6%	80.3%
VddWL=1.3V	100.0%	96.0%	83.8%

Figure 6. Yield Improvement.

4.2 Yield Enhancement

The parametric yield result for L1 data and instruction caches for the 1000 simulated chips are shown in Figure 6.

We simulated the latency of the top 128 critical paths of the L1 data and instruction caches under low, medium, and high process variations as described in the previous section. We assume the timing specification (T_{cutoff}) of the caches to be the mean latency ($\mu_{Tmedium}$) for the medium variation case plus half its standard deviation ($\sigma_{Tmedium}$). That is:

$$T_{cutoff} = \mu_{Tmedium} + 0.5\ \sigma_{Tmedium}.$$

Any cache with

$$Max(T_{pathi} \mid i=1 \text{ to } 128) > T_{cutoff}$$

is considered to be a yield loss. The same T_{cutoff} is used for all the three process variation scenarios. Hence, for a baseline cache without boosted wordline voltage, higher yield is seen when the process variations are low, while significant yield losses are experienced when the process variations are high. Across the variation scenarios, raising the wordline voltage can significantly improve the parametric yield: for the medium variation case, when the wordline voltage is raised to 1.0V, cache yield is improved by 10.8%, which corresponds to a 45.0% loss reduction. For the same variation levels, when the wordline voltage is raised to 1.3V, cache yield is improved by 28.5%, corresponding to an 85.2% reduction in the yield losses. For severe process variations, our scheme improves the yield level by up to 42.5%, corresponding to over 60% reduction in yield losses. For weaker process variations, our scheme can eliminate all the yield losses.

Based on our simulations across all the 1000 chips, 11.7% of the critical paths (rows) failed the timing requirement for the medium process variation scenario, on average. However, due to the spatial correlation in the process variations, the failing rows are not evenly distributed. Some of the failing chips have a larger number of failing rows, whereas others have few. Although redundancy schemes may save some of these chips, it alone will not be enough to improve the yield levels considerably particularly under severe process variations. However, our technique allows any number of wordlines to be raised to higher voltage, thus has the potential to improve the yield even with cases of massive failures. In fact, redundancy schemes can be used on top of our technique to recover particle related failures to further improve the overall yield.

4.3 Cost

The proposed design improves the cache access time at the price of increased energy and area consumption.

4.3.1 Energy consumption

In our proposed technique, we only raise the supply voltage of the wordline drivers for failing lines to minimize the energy overhead. Table 3 shows the normalized dynamic and static energy consumption for each scheme. Baseline scheme uses the nominal voltage (0.9V) for wordline voltage. Selective 1.0~1.3V schemes represent our proposed technique where only the wordline driver supply voltage is raised to the corresponding level. As a reference, we also show a case where the power supply voltage of the whole cache is raised to 1.0V (denoted by *Whole 1.0V*). The reported dynamic energy numbers correspond to reading a 64-bit word in a boosted line. For boosted lines, there is a small dynamic power overhead (2.1%~9.8%) compared with the baseline scheme as shown in Table 3. Note that only the originally failing lines face this overhead. Compared to raising the supply voltage of the whole cache to 1.0V, even the Selective 1.3V scheme consumes 80.4% less dynamic power. As described in Section 4.2, about 11.7% of the cache wordlines need to be boosted. Hence, the dynamic power consumption overhead in practice is even smaller (0.98% on average for the Selective 1.3V scheme). In addition, assuming 100 wordlines (10% of the complete set of wordlines) need to be boosted, we calculated the leakage overhead of our approach. The leakage overhead is also negligible (1.25% for

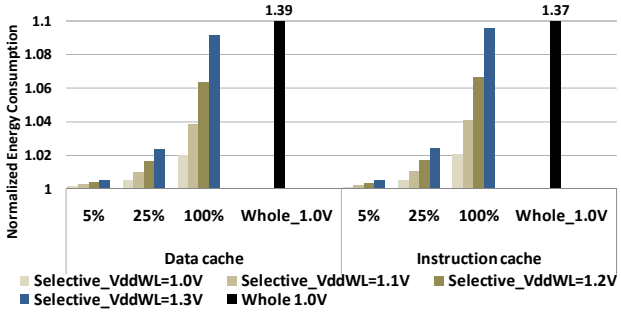


Figure 7. Geometric mean of normalized energy consumption in L1 caches across SPEC2K benchmarks. All of the schemes are normalized to baseline scheme using the nominal voltage.

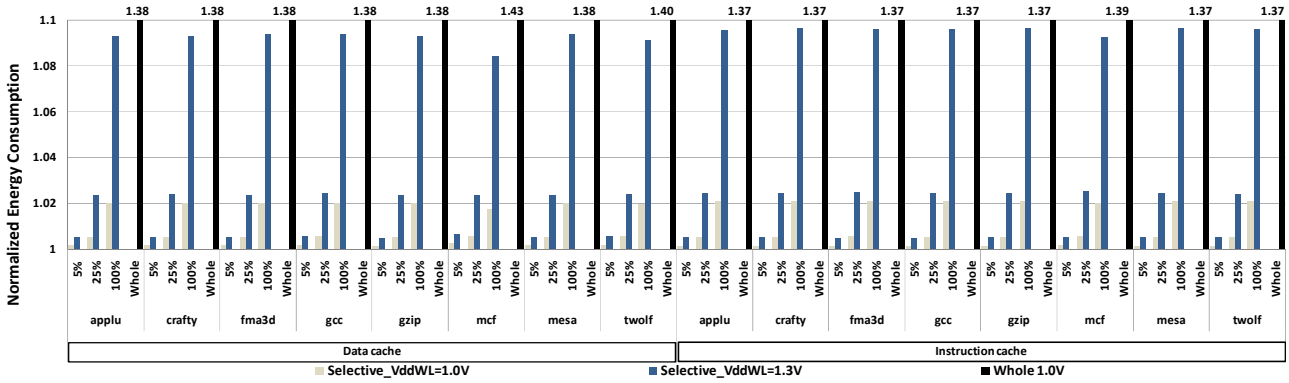


Figure 8. Normalized energy consumption in L1 caches when executing SPEC2K benchmark suite. All of the schemes are normalized to baseline scheme which uses the nominal voltage (0.9V).

Selective 1.3V) since the wordline leakage constitutes only a small fraction of the total leakage, while the sub-threshold leakage in the SRAM array dominates the cache leakage.

The actual power consumption impact varies depending on the chip sample as well as the application executed since the fault bitmap is different across chip samples and cache access patterns are different among applications. For example, there will not be any energy overhead if we do not access the boosted cache lines. Figure 7 shows the geometric mean of normalized energy consumption for the five schemes across the eight selected SPEC2000 applications. To gather these results, we generate 100 chips with randomly distributed failing cache rows. For example, with the 5% failure rate, we randomly select 5% of the rows for each of the 100 chips. Then, we simulate the eight applications monitoring the fraction of accesses that are made to failing rows. Using this information, we calculate the energy overhead of our proposed schemes. The x-axis shows the failure rate. A reference scheme with the whole cache power supply voltage raised to 1.0V is also shown (*Whole 1.0V*). In general, the dynamic power overhead is proportional to the number of boosted wordlines. For a case of 25% wordlines of data cache boosted to 1.0V and 1.3V, the geometric mean of the energy overhead is 0.55% and 2.3%, respectively. In the extreme case of having 100% boosted wordlines, the overhead can be as high as 9.2%. This is still drastically lower than that (39%) of raising the whole cache power supply to 1.0V. Instruction cache shows similar trends.

Figure 8 shows energy consumption of the L1 cache for specific applications. We present results for three schemes: "selective 1.0V", "selective 1.3V", and "whole 1.0V". As shown in Figure 8, "selective 1.0V" and "selective 1.3V" with 25% of the wordlines boosted has at most 0.58% and 2.3% energy overhead (for mcf application) compared to the baseline scheme, respectively. The studied applications show very similar behaviors. Overall, our selective schemes are much more energy-efficient, when compared to raising the supply voltage of the whole cache, regardless of which application is running on the microprocessor.

4.3.2 Area

Our proposed technique requires extra storage for the fault bits, inverters for logic generation, and power gating PMOS transistors. The power gating PMOS transistors take 3.54% of total L1 cache area. This overhead is comparable to similar techniques such as Gated-V_{dd} [22]. The storage needed for the fault bits is limited as only 128 Bytes (1024 bits) for a 32KB cache, which is less than 0.4% of the cache capacity. The total area cost can be conservatively estimated to be below 4.5% of total cache area.

Further area reduction can be achieved by increasing the granularity of control. By sharing the power gating PMOS across *n* wordlines, the area cost can be significantly reduced to one *n-th* of the above estimated value. This comes at the cost of extra power consumption for cases with sporadic failing lines.

5. RELATED WORK

There have been many studies on robust microprocessor design under process variations. Agarwal et al. [1][2] analyzed the cache operation failures under process variation and proposed process variation tolerant cache architecture, where the cache access failure is shown to be a dominant factor as σV_{th} is increased. Their proposed cache architecture utilizes remapping of column mux when there are faulty cache lines. Though their architecture improves yield significantly, it is not suitable for severe process variation environments because the reduced number of available cache lines leads to large performance overhead. The IBM Cell [24] and Power6 [9] processor both use increased array supply voltage to improve stability and read performance. However, their technique exercises a much coarser control granularity and they raise SRAM cell supply while we proposed to have fine-grain wordline voltage boosting. Chen et al. [6] compared three types of SRAM design under yield constraints. They compared these three types of SRAM designs (differential 6T, single-ended 6T, and 8T SRAM) in terms of energy and area considering transistor sizing for their robustness. Mutyam et al. [18] proposed process variation tolerant block rearrangement technique. This technique rearranges cache blocks which have similar access latency. It tries to minimize performance overhead incurred by process variation. However, their technique has performance overheads since their technique does not reduce the latency of faulty cache line but only allocates the cache lines that have similar latency in a same set through the block rearranging scheme.

In the circuit level, many techniques have been proposed to maintain the yield under process variation. Chen et al. [7] compared Adaptive Body Bias (ABB) to Adaptive Supply Voltage (ASV). Though both ABB and ASV can be used to reach sufficient yield, adopting either ABB or ASV is enough for yield improvement according to their analysis. They also concluded that ASV is simpler to implement than ABB. Tschanz et al. [36] introduced variation tolerant circuit techniques: ABB and adaptive V_{dd}. They suggested that using both ABB and adaptive V_{dd} is efficient with respect to variation tolerance. Li et al. [16] proposed

ASV technique which can supply V_{dd} adaptively in the processor pipeline. Though their technique reduces power consumption under the yield constraints, it is not applicable to caches where timing is critical as they confessed. Gregg et al. [10] proposed Individual Well Adaptive Body Biasing (IWABB). By adopting their circuitry, many biasing modes are available after fabrication considering the characteristics of each chip. Using their design flow adopting IWABB, yield can be significantly improved. However, their technique is based on ABB, which is more complicated to implement than adjusting V_{dd}. Teodorescu et al. [33] proposed Dynamic Fine-Grain Body Biasing (D-FGBB). Compared to conventional body biasing techniques that are statically applied to the fabricated chips, they proposed a dynamic body biasing technique which provides more flexibility. However, they do not exploit the special features of SRAM arrays.

6. CONCLUSION

As process technology scales down continuously and more transistors are packed in a limited die area, parametric yield losses have become an important problem for manufacturers. In this paper, we proposed a novel yield-aware selective wordline voltage boosting technique for caches. By boosting the voltage on selected wordlines, the overall yield of the cache can be significantly improved. SPICE simulations show over 80% reductions in yield losses, with average per-access energy overhead of less than 1%. The area overhead is well below 4.5% and can be further reduced by increasing the control granularity.

7. ACKNOWLEDGEMENT

This work was in part supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. R01-2007-000-20750-0); US National Science Foundation (NSF) grants CNS-0551639, IIS-0536994, CCF-0747201, and CCF-0541337; DoE CAREER Award DEFG02-05ER25691; and by Wissner-Slivka Chair funds.

8. REFERENCES

[1] A. Agarwal, et al., “A process-tolerant cache architecture for improved yield in nanoscale technologies”, *IEEE Transaction on VLSI Systems, vol. 13*, pp. 27-38, 2005.

[2] A. Agarwal, et al., "Process variation in embedded memories: failure analysis and variation aware architecture", *IEEE Jnl. of Solid-State Circuits, vol. 40*, 2005.

[3] S. Borkar, et al., “Parameter variations and impact on circuits and microarchitecture”. *In Proc. of DAC*, 2003.

[4] K. A. Bowman, et al., “Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration”. *IEEE Jnl. of Solid-State Circuits, vol. 37*, 2002.

[5] W. Bryg and J. Alabado, "The UltraSPARC T1 Processor - Reliability, Availability, and Serviceability."

[6] G. K. Chen, et al., “Yield-driven near-threshold SRAM design”. *In Proc. of Int’l Conference on Computer Aided Design*, 2007.

[7] T. Chen and S. Naffziger. "Comparison of Adaptive Body Bias (ABB) and Adaptive Supply Voltage (ASV) for Improving Delay and Leakage under the Presence of Process Variation". *IEEE Transaction on VLSI Systems, vol. 11*, pp. 888-899, 2003.

[8] A. Das, S. Ozdemir, G. Memik, J. Zambreno, and A. N. Choudhary, “Microarchitectures for Managing Chip Revenues under Process Variations”. *Computer Architecture Letters, vol. 6*, pp. 29-32, 2007.

[9] J. Friedrich, et al., “Design of the Power6 Microprocessor”, In ISSCC, 2007.

[10] J. Gregg and T. W. Chen. “Post Silicon Power/Performance Optimization in the Presence of Process Variations Using Individual Well Adaptive Body Biasing (IWABB)”. *In proceedings of IEEE Int’l Symposium on Quality Electronic Design*, 2007.

[11] F. Hamzaoglu, et al., "A 153Mb-SRAM Design with Dynamic Stability Enhancement and Leakage Reduction in 45nm High-k Metal-Gate CMOS Technology", *In Proc. of Int’l Solid State Circuits Conference*, 2008.

[12] N. P. Jouppi, "Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers", *ACM SIGARCH Computer Architecture News, vol. 18*, 1990.

[13] C. Kim, D. Burger, and S. W. Keckler. “An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches”. *In Proc. of Int’l Conference on Architectural Support for Programming Languages and Operating Systems*, 2002.

[14] N. Kim, et al., “Single-VDD and single-VT super-drowsy techniques for low-leakage high-performance instruction caches”, *In Proc. of Int’l Symposium on Low Power Electronics and Design*, 2004.

[15] K. Krewell, Alpha EV7 Processor: A High-Performance Tradition Continues, Apr. 2002.

[16] H. Li, et al., “SAVS: a self-adaptive variable supply-voltage technique for process- tolerant and power-efficient multi-issue superscalar processor design”. *In Proc. of ASPDAC*, 2006.

[17] X. Liang, et al., “Process Variation Tolerant 3T1D-Based Cache Architectures”. *In Proc. of IEEE/ACM Int’l Symposium on Microarchitecture*, 2007.

[18] M. Mutyam and N. Vijaykrishnan. “Working with process variation aware caches”. *In proc. of DATE*, 2007.

[19] S. Naffziger, et al., "The Implementation of the Itanium 2 Microprocessor," *IEEE Jnl. of Solid-State Circuits, vol. 37*, 2002.

[20] S. R. Nassif. “Modeling and Analysis of Manufacturing Variations”. *In Proc. of IEEE Conf. on Custom Integrated Circuits*, 2001.

[21] S. Ozdemir, D. Sinha, G. Memik, J. Adams, and H. Zhou, “Yield-Aware Cache Architectures”. *In Proc. of IEEE/ACM Int’l Symposium on Microarchitecture*, pp. 15-25, 2006.

[22] M. Powell, et al., “Gated-Vdd: A circuit technique to reduce leakage in deep-submicron cache memories”, ISLPED, pp. 90-95, 2000.

[23] Predictive Technology Model (PTM), Arizona State University. http://www.eas.asu.edu/~ptm/

[24] M. Riley, et al., “Implementation of the 65nm Cell Broadband Engine”, *In Proc. of IEEE Custom Integrated Circuits Conf.*, 2007.

[25] B. F. Romanescu, et al., "Reducing the Impact of Intra-Core Process Variability with Criticality-Based Resource Allocation and Prefetching". *In Proc. of ACM Int’l Conference on Computing Frontiers*, 2008.

[26] T. Sakurai and A. R. Newton. “Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas”, *IEEE Jnl. of Solid-State Circuits, vol. 25*, pp. 584-594, 1990.

[27] S. R. Sarangi, et al., “VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects”, *IEEE Transactions on Semiconductor Manufacturing, vol. 21*, pp. 3-13, 2008.

[28] T. Sherwood, et al., “Automatically characterizing large scale program behavior”. *In Proc. of Int’l Conference on Architectural Support for Programming Languages and Operating Systems*, 2002.

[29] P. Shivakumar, et al., "Exploiting microarchitectural redundancy for defect tolerance". *In Proc. of IEEE Int’l Conference on Computer Design*, 2003.

[30] SIA. International Technology Roadmap for Semiconductors, 2005. Available at http://public.itrs.net.

[31] SimpleScalar toolset. http://www.simplescalar.com.

[32] G. S. Sohi, "Cache Memory Organization to Enhance the Yield of High Performance VLSI Processors". *IEEE Transaction on Computers, vol. 38*, pp. 484-492, 1989.

[33] R. Teodorescu, et al., “Mitigating Parameter Variation with Dynamic Fine-Grain Body Biasing”. *In Proc. of IEEE/ACM Int’l Symposium on Microarchitecture*, pp. 27-42, 2007.

[34] The R Project for Statistical Computing. Available from: http://www.r-project.org.

[35] S. Thoziyoor, et al., “CACTI 5 Technical Report. HP Labs”.

[36] J. Tschanz, K. A. Bowman, and V. De. “Variation-tolerant circuits: circuit solutions and techniques”. *In Proc. of Design Automation Conference*, pp. 762-763, 2005.

[37] B. Zhai, et al., “The limit of dynamic voltage scaling and insomniac dynamic voltage scaling”, *IEEE Transaction on VLSI Systems, vol. 13*, pp. 1239-1252, 2005