Replacing 6T SRAMs with 3T1D DRAMs in the L1 Data Cache to Combat Process Variability

With continued technology scaling, process variations will be especially detrimental to six-transistor static memory structures (6T SRAMs). A memory architecture using three-transistor, one-diode DRAM (3T1D) cells in the L1 data cache tolerates wide process variations with little performance degradation, making it a promising choice for on-chip cache structures for next-generation microprocessors.

Xiaoyao Liang
Harvard University

Ramon Canal
Universitat Politècnica de Catalunya

Gu-Yeon Wei
David Brooks
Harvard University

Technology nanoscaling promises increasing transistor density and increasing performance in microprocessors. However, the road toward this promise is fraught with difficulties resulting from increased process variations that limit performance gains and affect the stability of key circuit blocks such as on-chip memories. Addressing these problems requires innovation at all levels in the design flow—from devices to system architecture.

Fluctuations in device channel dimensions and dopant concentrations are two forms of process variation that affect circuit performance. Gate length variation can change the transistor's effective driving capability and the threshold voltage due to short channel effects. Random dopant variations also change each device's threshold voltage. The semiconductor manufacturing process gives rise to both within-die variations (that is, varying device features on one chip) and die-to-die variations (varying device features on different chips). As technology scales, within-die variations are getting wider, significantly affecting performance and compromising circuit reliability.

In modern microprocessors, on-chip memories consume a significant portion of overall die space, providing high system performance (in contrast to slow off-chip memories) in exchange for the space and power they consume. On-chip caches traditionally rely on SRAM cells, which have generally scaled well with technology. Because of increasing device-to-device mismatch and variation, however, stability, performance, and leakage power will become major hurdles for the continued scaling of SRAMs implemented in aggressive nanoscale technologies. To avoid SRAM scaling limitations, we need new circuit and architectural solutions.

Given the transient nature of data flow, dynamic memories are good candidates for data storage structures in the processor core. We propose using dynamic memory cells with architectural support to enable robust

Published by the IEEE Computer Society

cache designs that are tolerant to process variations for future generations of high-performance microprocessors. Specifically, we propose an on-chip memory architecture based on three-transistor, one-diode (3T1D) dynamic memory cells.[1] In this article, we demonstrate a robust memory design in an important latency-critical component of the processor—the L1 data cache—identified as one of the architectural components most susceptible to process variations.[2,3] Architectural policies can mitigate the costs associated with dynamic memories and overcome effects of physical device variations, while offering significant performance, stability, and power advantages. Compared with traditional SRAM designs, the proposed design makes it easier to apply fine-grained variation control, which is very important for designs that suffer large within-die process variations.

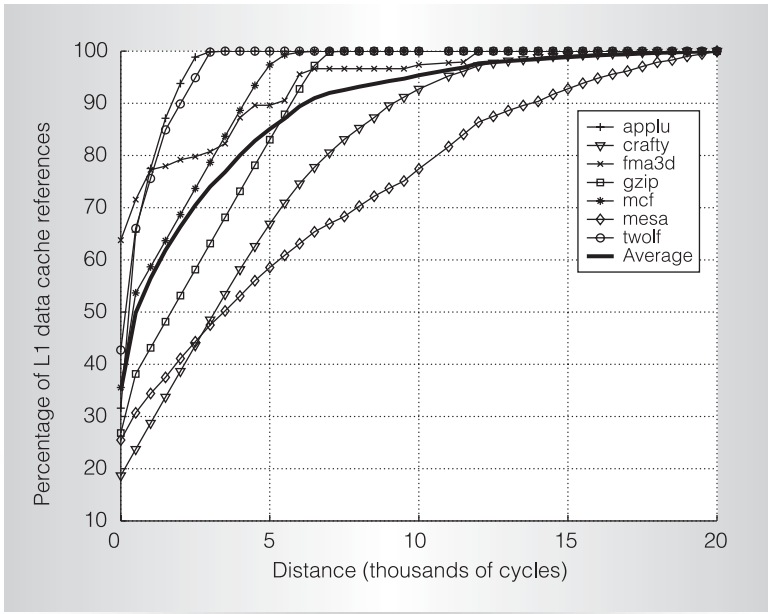

Figure 1. Percentage of L1 data cache references versus elapsed clock cycles.

Overview

For decades, the six-transistor (6T) SRAM has been the de facto choice for on-chip memory structures, such as register files and caches, in the processor core. Designers have used DRAM memory technologies for larger off-chip caches or main memories. These choices have been driven primarily by the high memory access speed provided by SRAMs and the high density provided by DRAMs. Because its memory storage node is a capacitor, a DRAM stores data temporarily and requires periodic refresh if it must hold data for extended periods.

On the other hand, a large fraction of data consumed in the processor core is transient.[4] As Figure 1 shows, most cache accesses happen within the initial 6,000 clock cycles after the data is loaded. If a dynamic memory's data retention time meets system requirements, it is a better choice for on-chip temporary data storage because the temporal characteristics of the data in the processor can reduce or even eliminate the need for refresh.

Circuit imbalances due to mismatch can compromise cell stability and exacerbate leakage in traditional SRAM designs. Furthermore, the many critical paths and the few devices in each path contribute to increased access time variability. This variability is especially important for higher levels of the memory hierarchy, which are more latency critical. In contrast, although lower memory levels (such as the L2 cache) are less sensitive to latency, they are susceptible to bit flips. The 6T cell's reliance on symmetry and careful sizing requirements to accommodate robust reads and writes makes it susceptible to random variations.[5] One simple solution to these problems is to slow down SRAM scaling at the expense of lower performance and larger area. However, this would effectively mean the end of Moore's law for transistor density and speed scaling for future processors. In light of these 6T SRAM scaling concerns, architectural support is necessary to enable the use of dynamic memories for on-chip L1 data caches.

Recent circuit innovations in memory design provide an interesting alternative to 6T cells. Luk et al. proposed a novel 3T1D DRAM cell, which offers speed comparable to a 6T SRAM cell for a limited time after writing the data.[1] Published results on chips fabricated in 130-nm and 90-nm technologies verify high-speed operation for these 3T1D memories. The 3T1D cells are well

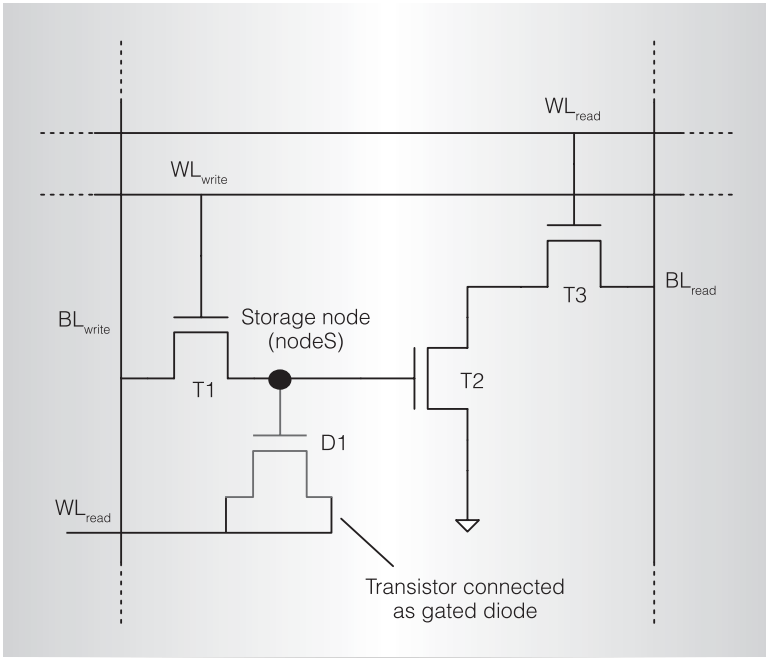

Figure 2. Schematic of 3T1D DRAM cell. WL: wordline; BL: bitline.

suited for latency-critical structures, whereas 1T cells might be more appropriate for slower L2 caches. The memory architecture we propose will scale more effectively with technology under increasing process variations by employing intelligent data retention schemes for dynamic memories.

Figure 2 presents a schematic of the 3T1D DRAM cell. Its read path resembles that of a 6T SRAM cell but relies on a voltage-controlled capacitor (D1) to selectively boost the stored voltage (when reading a 1) and overcome the degraded level caused by T1's threshold voltage. As a result, the access speed can exceed the speed of 6T SRAM cells. Conversely, when the cell stores a 0, D1's capacitance is smaller and there is almost no voltage boosting, which keeps T2 off during the read.

Although a 3T1D cell can be fast, high-speed access is valid only for a limited period after each write to the cell because charge leaks away over time. Figure 3 plots the relationship between cell access time and elapsed time after a write operation. Access time degrades until finally exceeding the

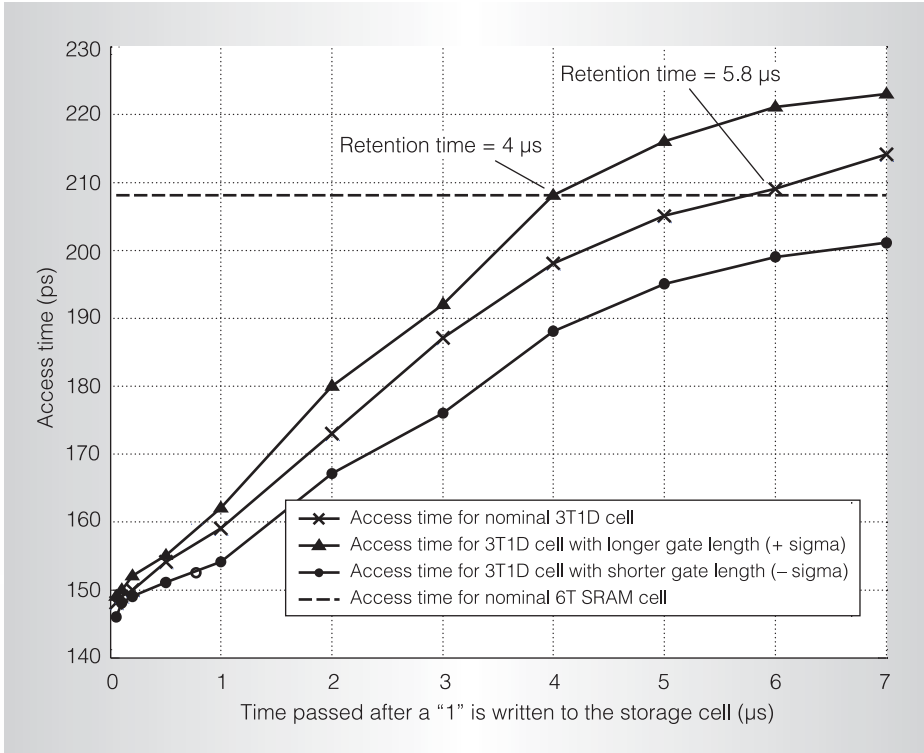

Figure 3. Relationship between access time and data retention time in a 3T1D cell.

array access time of a 6T SRAM. We define a 3T1D memory's retention time as the period during which the access speed can match that of a 6T SRAM cell (about 5.8 μs for nominal cells in Figure 3). Within this retention time, the memory can be accessed at the nominal chip frequency. After this retention time passes, the memory cell must be refreshed (rewritten); otherwise, that cell is considered invalid.

A 3T1D memory is not entirely immune to process variation, but a single parameter can absorb its effects and deal with it efficiently. For example, if process variation reduces access transistor T3's driving capability, access time increases. This effect can otherwise be viewed as decreasing retention time, as shown in Figure 3. Weaker-than-designed access transistors have the effect of shifting the access time curve to the left, reducing cell retention time. Process variation does not necessarily impact operating frequency, which is the case for 6T SRAMs. In short, process variation's impact on a 3T1D memory can all be lumped into a single variable—retention time. Furthermore, 3T1D cells don't suffer the same type of cell stability issues as do 6T SRAM cells, because there is no inherent fighting in 3T1D cells. Except for the finite data retention time, a 3T1D DRAM cell is stable. The 3T1D-based memory array also offers advantages in reduced leakage power because a 3T1D cell doesn't suffer the multitude of strong leakage paths between V_{DD} and ground found in 6T SRAM cells. This smaller number of leakage paths leads to lower nominal leakage and less leakage variability.

Process-variation-tolerant cache architectures

What kind of architectural support is needed for 3T1D cells to replace traditional 6T cells? Several approaches accommodate the limited retention time of 3T1D memories: periodic data refreshing, retention-time-driven replacement policies, allowing data to expire (no refresh), and combinations of these approaches.

We performed our experiments with a superscalar, out-of-order processor with a 64-Kbyte, 512-bit block size, 4-way set-associative L1 data cache. We constructed circuit-level models for the cache with the 32-nm predictive technology model (PTM) library and performed Monte Carlo simulations to study variability for both 3T and 6T caches. We considered two process variation scenarios: one with typical variations and the other with severe variations. We first studied the 3T1D cache without considering any variations and then investigated several variation-tolerant strategies.

3T1D cache without process variation

The simplest way to provide support for data retention in 3T1D caches is to use a global refresh scheme. Refresh operations require a read and a subsequent write-back to the memory. As Figure 4a shows, when the cache needs a refresh, the processor blocks one read and one write port from normal cache accesses and uses them to refresh the data. The refresh mechanism pipelines these read and write accesses on consecutive cycles. This approach introduces a performance penalty because the refresh operation competes with normal instructions. It takes 8 clock cycles to refresh one cache line, thus requiring 2,048 cycles to complete the refresh. In our 32-nm 4.3-GHz machine, this requires 476.3 ns, using about 8 percent of cache bandwidth, which leads to less than 1 percent performance loss.

Dealing with typical variation: Global-refresh scheme

Process variations affect the operating frequency of 6T SRAM caches. In contrast, process variations introduce retention time variations in 3T1D caches. For each fabricated cache, the memory cell with the shortest retention time determines the entire structure's retention time. Figure 5a presents the chip distribution (or probability of manufacturing) as a histogram of retention time distribution for the 3T1D cache with typical process variations. Although all the chips are designed with the same parameters, they exhibit a wide retention time spread due to process variations. However, Figure 5b shows that processor performance only varies by about 2 percent, with retention time varying from 714 ns to 3,094 ns. In Figure 5b, 97 percent of the

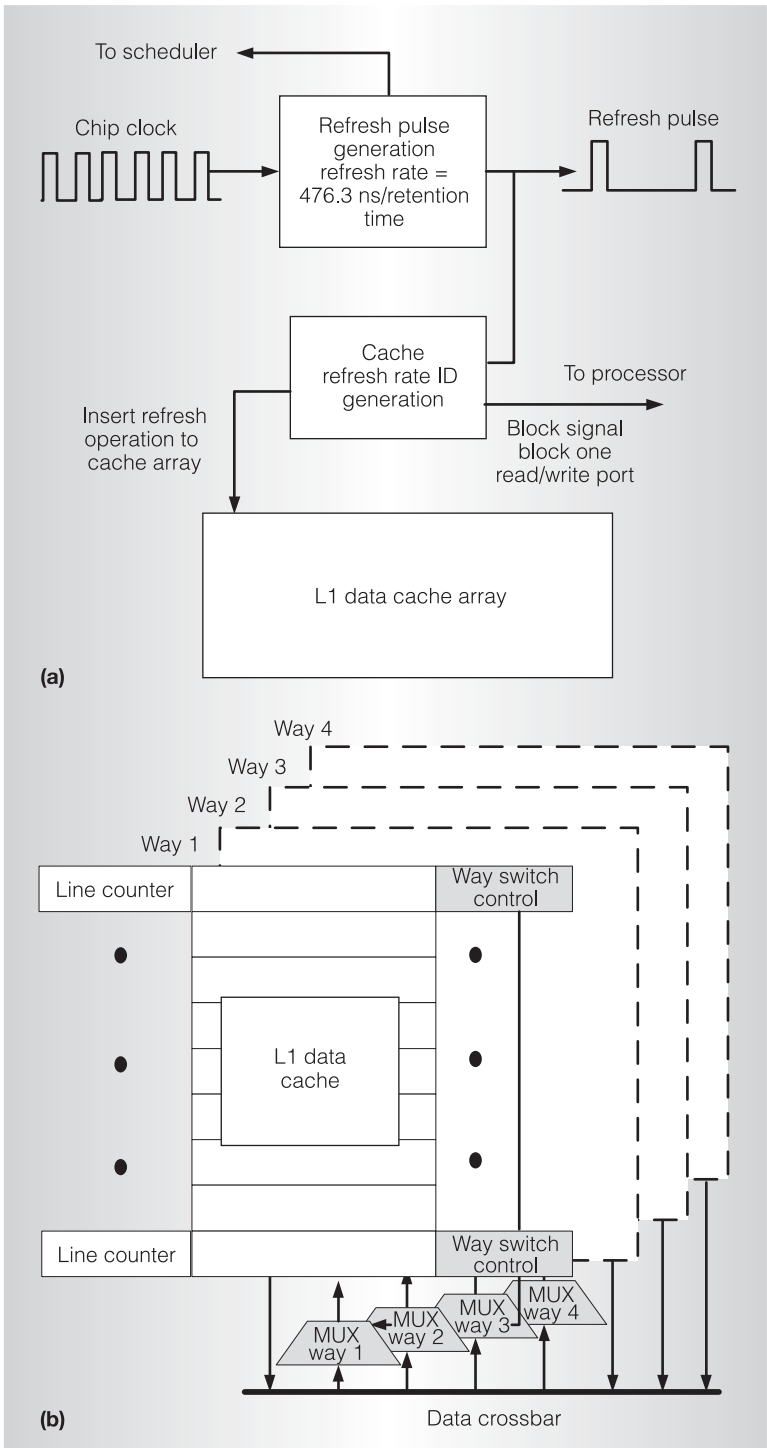

Figure 4. Proposed cache architectures for global (a) and line-level (b) schemes. Shaded blocks are used only in the retention-sensitive-placement-FIFO (RSP-FIFO) and least-recently-used (RSP-LRU) schemes.

3T1D caches lose less than 2 percent of performance compared with an ideal 6T design. Even for the worst-performing benchmark (fma3d), the performance penalty is less than 4 percent. Therefore, a 3T1D cache achieves much better performance than a SRAM design for comparable yields. However, this increased performance comes at the expense of additional dynamic energy consumption for refresh. Dynamic-power overhead for the 3T1D ranges from 1.3 to 2.25 times that of the ideal 6T design, as Figure 5c shows. Although dynamic power consumption is larger, leakage power tends to dominate cache structures at 32 nm, and as a result there is an overall reduction in cache power compared with the SRAM design.

Dealing with severe variation: Line-level schemes

A major disadvantage of the 6T SRAM cache is that the speed of its slowest cells determines the entire structure's operating frequency. The global-refresh scheme for the 3T1D cache has a similar problem—the worst memory cell determines the entire structure's retention time. Although the global-refresh scheme works well for typical process variations, two problems make it perform poorly under severe process variations. First, under severe variations, some cache lines cannot function properly (retention times are too low); we call these "dead lines." Because the entire cache's retention time in the global scheme is limited by the worst cache lines, chips with dead lines must be discarded. This inefficiency arises because the global scheme is too coarse-grained and cannot distinguish the differing retention time requirements of each cache line. Explicit global refresh also introduces significant dynamic-power overhead to the system. To solve these two problems, we propose retention schemes that operate at the line level. These schemes allow the operating frequency to remain constant for 3T1D memories, and architectural policies manage variable retention times.

Refresh policies. Line-level refresh requires each line to be tagged with a retention time (defined by the lowest retention time of the cells in that line, so that no data is lost

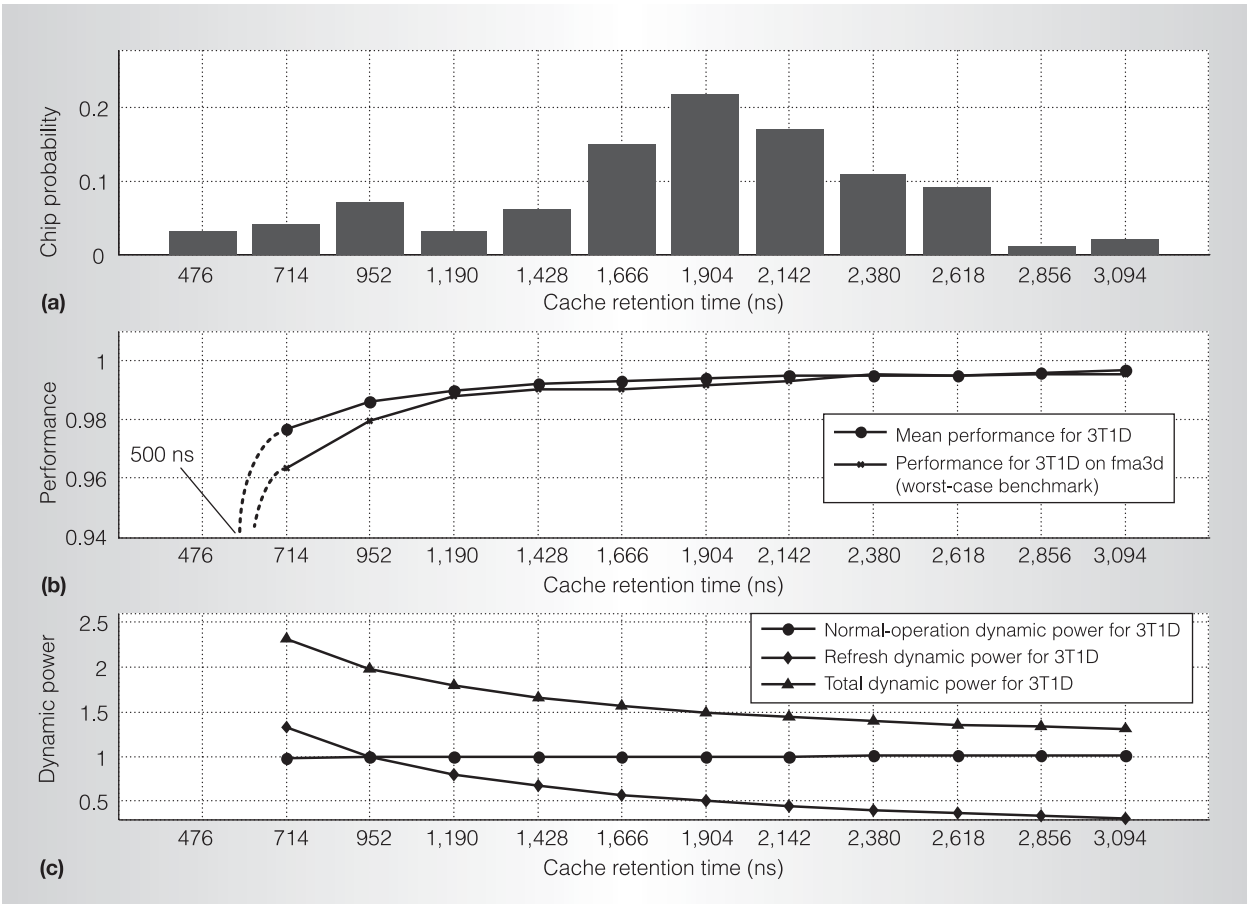

Figure 5. Retention time distribution (a), performance (b), and power (c) for the 3T1D cache. Performance and power are presented relative to an ideal 6T cell.

during the retention time). In terms of hardware budget, line-level refresh requires an extra counter per line to individually keep track of retention times. The line counter is reset when a new data block is loaded to the cache line and starts counting. Once it counts to the line retention time, that line is no longer useful and must be refreshed or evicted. All line counters are synchronized to a global clock, which is a fraction (1/*N*) of the chip frequency. This means the line counter's minimal time step is *N* cycles. *N* can be set according to variation conditions.

The spectrum of possible refresh policies ranges from refreshing each individual line to completely avoiding refresh and relying on the memory hierarchy's inclusion properties to recover data from the L2. We have identified three schemes in the spectrum:

- *No refresh.* This scheme refreshes no cache lines. Thus, whenever a cache line's retention time reaches a low threshold, the line is evicted. Although write-through caches require no action, write-back caches require dirty data to be written to the L2 cache. In the pathological scenario of many dirty lines expiring simultaneously, write-back traffic can stall the write buffer. To ensure data integrity, dirty lines waiting for eviction are refreshed during this stall. The only hardware overheads are line counters and control logic.
- *Partial refresh.* This technique tracks each line's retention time. If the retention time is below a specific threshold, that line is refreshed, and it stays alive until passing the threshold

time. Any lines with retention times larger than the threshold are not refreshed. Thus, only a subset of lines is refreshed, depending on their distinct retention times. This scheme guarantees that all lines have a lifetime longer than the threshold before they are evicted. To ensure data integrity, we conservatively set the retention time counter to guarantee that each line requesting refresh receives a refresh before expiring.

- *Full refresh.* All cache lines are always refreshed before the retention time expires. This scheme is similar to the partial-refresh scheme. The only difference is that all cache lines are refreshed, and there is no threshold time.

Replacement policies. The cache line replacement policy also provides an important approach to supporting the 3T1D cache's data retention requirements. Conventional least-recently-used (LRU) policies always select the least-used block for replacement, which may not be the best choice for 3T1D caches. Under process variations, lines in each set can have different retention times. As the line usage of the benchmarks in Figure 1 shows, most (90 percent) accesses to a line are concentrated in its first 6,000 cycles of lifetime. Thus, it is usually not effective to assign newly loaded data to the way with the lowest retention time, because the cache might not retain the data for sufficient time. Thus, we propose three retention-time-sensitive replacement policies:

- *Dead-sensitive placement (DSP).* Some cache lines have zero retention time and are dead because of within-die variations. We also treat a cache line as dead if its retention time is less than the line counter's minimal time step (N). This policy replaces lines in a manner that avoids using the dead lines. In replacing lines, it handles them in a manner similar to a conventional LRU scheme but uses no dead lines. For example, in a 4-way cache with one dead line, the replacement policy always uses the other three ways. In the worst case, given a set with all ways dead, the accesses or replacements miss the L1 cache and incur a corresponding L2 access.
- *Retention-sensitive-placement FIFO (RSP-FIFO).* This policy logically organizes the lines in each set in a FIFO scheme ordered by descending retention times. Thus, new blocks are assigned the line with the longest retention time, and blocks in that set move to the line with the next-longest retention time. Data held in the line with the lowest retention time is evicted. This policy relies on the theory that most cache accesses occur within the initial period after the data is loaded. Moving newly loaded data into longer-retention-time locations can greatly reduce the number of cache misses resulting from expired retention times. This mechanism intrinsically incorporates a refresh since every new block inserted into the cache causes the existing blocks in that set to move to new lines (with smaller retention times). The scheme requires multiplexers to switch data between the different ways, as the shaded blocks in Figure 4b show.
- *Retention-sensitive-placement-LRU (RSP-LRU).* This policy is similar to the previous one, but the most recently accessed block remains in the line with the longest retention time. Thus, every read or write (hit or miss) can shuffle lines among the ways to keep them in the appropriate order (highly active blocks reside in the higher-retention-time lines). The policy relies on the theory that data accessed in the cache will likely be accessed again soon (temporal locality). The policy is complex because the line shuffling occurs more frequently than in RSP-FIFO, which shuffles lines only on cache evictions.

Evaluating the schemes. The cross-product of the three refresh schemes and the four replacement policies (including conventional LRU) yields eight viable retention time

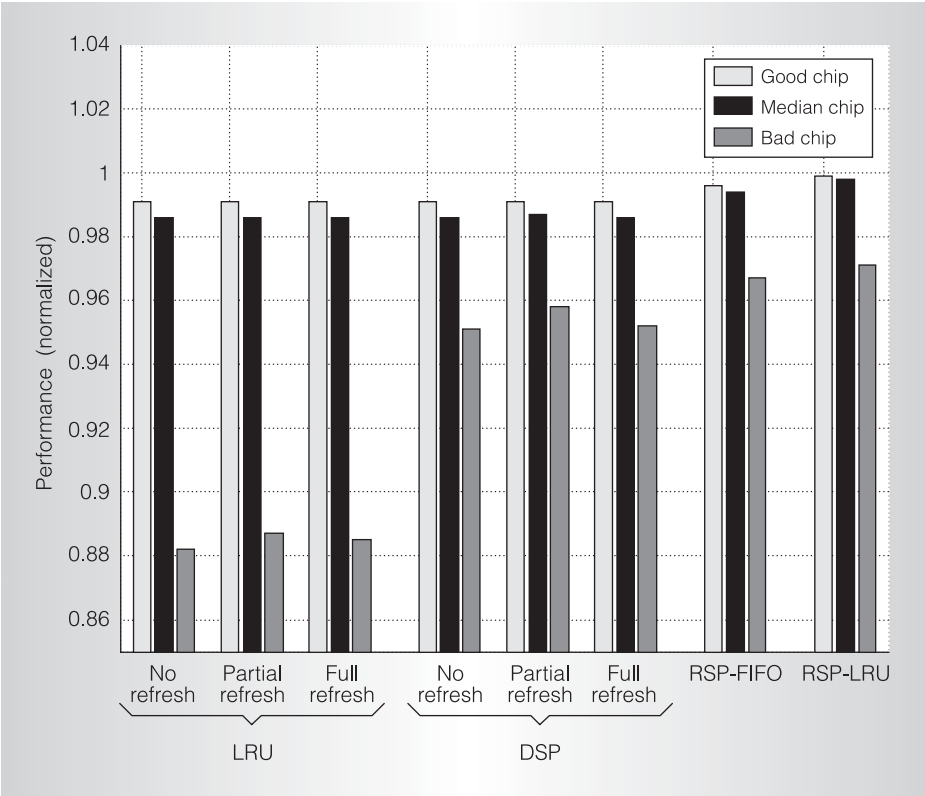

Figure 6. Normalized performance (compared with an ideal 6T design) of retention schemes for good, median, and bad chips.

strategies to consider. We generated 100 sample chips under severe process variations. To evaluate the schemes' efficiency, we picked one good chip (representing a chip with process corners leading to very high retention times), one median chip, and one bad chip (very low retention times) in terms of process variation. We analyzed these 3T1D L1 cache based chips' performance for the eight schemes and plotted the performance relative to an ideal 6T cache design. Figure 6 summarizes the results.

For the bad chip, the differences between schemes are most prominent, and RSP-LRU can achieve performance within 3 percent of an ideal 6T memory (with no variations). These results demonstrate that line-level retention policies can overcome the large number of variations in the memory. Compared with the global-refresh scheme, which has 30 to 125 percent dynamic-power overhead, all the line-level schemes shown here are far more efficient in terms of dynamic power and leakage power benefits. Given these results, we recommend partial-refresh DSP as the most complexity-effective retention scheme.

In contrast, we would have to discard 80 percent of the chips when using the global scheme with the same number of variations, and the 6T cache would suffer a 40 percent frequency loss. Furthermore, almost every cache line in the conventional 6T cache would contain unstable cells. This amount of instability would be difficult to overcome with redundancy and error-correcting code mechanisms.

By leveraging underutilization of architectural resources, transient data characteristics, and program behavior, we have demonstrated significant performance and power benefits with 3T1D memories. Under severe variations, the schemes presented here perform with less than 4 percent performance loss and offer the inherent

extra benefit of reducing leakage power. These promising results suggest that 3T1D memories can replace existing on-chip 6T memories as a comprehensive solution to deal with process variations. MICRO

Acknowledgments

This work was supported by NSF grants CCF-0429782 and CCF-0702344, the Fulbright program, the Generalitat de Catalunya (2005SGR00950), and the Ministerio de Educación y Ciencia (TIN2007-61763).

References

1. W.K. Luk et al., ''A 3-Transistor DRAM Cell with Gated Diode for Enhanced Speed and Retention Time,'' *Proc. Symp. VLSI Circuits*, IEEE Press, 2006, pp. 184-185.
2. A. Agarwal et al., ''A Process-Tolerant Cache Architecture for Improved Yield in Nanoscale Technologies,'' *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 1, Jan. 2005, pp. 27-38.
3. S. Ozdemir et al., ''Yield-Aware Cache Architectures,'' *Proc. 39th Ann. IEEE/ACM Int'l Symp. Microarchitecture* (MICRO 06), IEEE CS Press, 2006, pp. 15-25.
4. D.A. Wood, M.D. Hill, and R.E. Kessler, ''A Model for Estimating Trace-Sample Miss Ratios,'' *Proc. ACM Sigmetrics Conf. Measurement and Modeling of Computer Systems*, ACM Press, 1991, pp. 79-89.
5. A.J. Bhavnagarwala, X. Tang, and J.D. Meindl, ''The Impact of Intrinsic Device Fluctuations on CMOS SRAM Cell Stability,'' *IEEE J. Solid-State Circuits*, vol. 36, no. 4, Apr. 2001, pp. 658-665.

Xiaoyao Liang is pursuing a PhD in electrical engineering at Harvard University. His research interests include computer architecture and VLSI design, currently focusing on joint circuit and architecture solutions to combat process variability. Liang has an MS from Stony Brook University and a BS from Fudan University, China, both in electrical engineering.

Ramon Canal is an associate professor in the Computer Architecture Department of the Universitat Politècnica de Catalunya (UPC), Spain. His research interests include power- and thermal-aware architectures, chip multiprocessors, and reliability. Canal has BS, MS, and PhD degrees in computer engineering from UPC. He was a Fulbright scholar at Harvard University.

Gu-Yeon Wei is an associate professor of electrical engineering at Harvard University. His research interests include mixed-signal VLSI design for wire-line data communication, energy-efficient computing devices for sensor networks, and collaborative software, architecture, and circuit techniques to combat variability in nanoscale technologies. Wei has BS, MS, and PhD degrees in electrical engineering from Stanford University.

David Brooks is an associate professor of computer science at Harvard University. His research interests include architectural and software approaches to address power, thermal, and reliability issues for embedded and high-performance computing systems. Brooks has a PhD in electrical engineering from Princeton University.

Direct questions and comments about this article to David Brooks, School of Engineering and Applied Sciences, Harvard University, 33 Oxford St., Cambridge, MA 02138; dbrooks@eecs.harvard.edu.