Process Variation Tolerant Register Files Based On Dynamic Memories

Xiaoyao Liang*, Ramon Canal†, Gu-Yeon Wei* and David M. Brooks*
*School of Engineering and Applied Sciences, Harvard University, Cambridge, MA
{xliang,wei,dbrooks}@eecs. harvard.edu
†Dept. of Computer Architecture, Universitat Politecnica de Catalunya, Cr. Jordi Girona 1-3, 08034 Barcelona, Spain
rcanal@ac.upc.edu

***Abstract*— Transistor gate length and threshold voltage variability due to process variations will greatly impact the stability, leakage power, and performance of future microprocessors. These variations are especially detrimental to continued scaling of 6T SRAM (6-transistor static memory) structures. This paper proposes replacing traditional SRAM-based cells in mutliported register files with cells based on 3T1D DRAM (3-transistor, 1-diode dynamic memory) cells, which can absorb the effects of device physical variations into a single parameter – the data retention time. By leveraging the transient data in the processor and dependency slack in the pipeline, retention time variation can be hidden into the existing processor architecture. Thus the proposed register file can effectively tolerate very large process variation with little or even no impact on performance, addresses stability concerns, and reduces power consumption, when compared with ideal SRAM-based designs. Detailed circuit and architectural simulations and analysis verify a 1% normalized performance loss even under very large process variations, and 22% average power savings.**

I. Introduction

Nanoscale technology scaling offers the promise of continuing transistor density and performance trends. However, the road is fraught with difficulties resulting from increased process variations that limit performance gains and affect stability of key circuit blocks such as on-chip memories. Addressing these problems will require innovation from all levels in the design flow from the devices to the system architecture. This paper investigates using dynamic memory cells with architecture support to enable robust register file designs tolerant to process variations for future generations of high-performance microprocessors.

Process variation is mainly caused by fluctuations in dopant concentrations and device channel dimensions. Gate length variation can change the effective driving capability of the transistor, as well as the threshold voltage due to the short channel effect. Random dopant variations can also change the threshold voltage of the device. The nature of the semiconductor manufacturing process gives rise to both within-die variations (i.e. device features on one chip can be different) and die-to-die variations (i.e. device features across chips can be different). As technology scales, within-die variations are getting larger, significantly affecting performance and compromising circuit reliability.

On-chip memories consume a significant portion of the overall die space in modern microprocessors due to their area efficiency and the high system performance they offer in exchange for the space and power they consume. Multi-ported register files rely on SRAM cells, which have generally scaled well with technology. Unfortunately, stability, performance, and leakage power will become major hurdles for future SRAMs implemented in aggressive nanoscale technologies due to increasing device-to-device mismatch and variations. One simple solution to these problems that affect traditional SRAM designs is to stop scaling SRAMs at the expense of lower performance and larger area. However, this would effectively mean the end of Moore's Law scaling of transistor density and speed for future processor designs.

To avoid these scaling limitations, new circuit and architectural solutions are needed. In this paper, we investigate on-chip memory architectures based on 3T1D dynamic memory cells [14]. We demonstrate a robust memory design in the context of an important architectural structure – multiported register files. We show how characteristics of the modern processor architecture (e.g., superscalar, out-of-order execution) can mitigate the costs associated with dynamic memories and overcome effects of physical device variations, while offering significant performance, stability, and power advantages. This paper takes several steps in the direction of variation-tolerant memory architecture designs. Specifically, there are three major contributions:

- We propose to use a 3T1D-based dynamic memory cell as the base for a new generation of register file designs. Dynamic memories can be a good candidate for data storage structures within the processor core, given the transient nature of data flow.
- The proposed register file can tolerate very large process variations with small or even no impact on performance. Effects of process variations can be absorbed into a single parameter – data retention time – efficiently addressed by simple architectural solutions.
- Besides performance benefits, the proposed register files are robust to memory cell stability issues and can achieve large power savings.

In the following section, we discuss background information and related work. Section III then investigates how process variations affect traditional 6T SRAM designs and provides a detailed comparison to 3T1D DRAMs. Section IV describes

the circuit and architecture simulation methodology used for the analysis presented in Section V, which shows how dynamic memories can lead to process variations tolerate register file designs. Finally, this work is summarized in Section VI.

II. Related Work

In recent years, process variation has been identified as one of the key threats to continued Moore's Law scaling, with projections that a technology generation of performance can be lost due to process variation [5] and serious concerns about the continued scalability of SRAM-based memories [3]. Circuit-level solutions to process variability include adaptive body biasing (ABB) to mitigate the impact of variation on digital logic [19], [26]. Several groups propose solutions to patch stability issues due to process variation in memory designs that use 6T SRAM cells [10], [22].

Recently, researchers have begun to explore the system-level impact of variations on power, performance, and reliability. Initial work in this area has focused on the modeling of process variations [9], [23]. Researchers have shown that the selection of pipeline depth [4], [11] and other microarchitectural parameters [12], at design time, can significantly impact the susceptibility of an architecture to process variations. Variable-latency techniques have been proposed for caches, register files, and pipelined logic structures [13], [20], [25]. Globally-asynchronous, locally synchronous (GALS) design techniques may offer ways to mitigate the impact of correlated within-die variations [17]. Agarwal et al. propose to resize caches in response to variation-induced failures after fabrication [2]. Process variations are also expected to have a substantial impact on leakage power, and researchers have begun to explore this problem in the context of caches [18] and multicore processors [8].

In contrast to the related works, we propose to replace on-chip SRAM with 3T1D DRAM memories. The proposed memory architectures offer advantages in terms of cell stability, reduced power requirements, and the ability to tolerate performance variations by intelligently tuning the refresh policies. This approach provides a comprehensive solution to many of the issues that will impact on-chip memory designs in nanoscale process technologies.

III. Comparison Between 6T Static Memory and 3T1D Dynamic Memory

For years, the traditional 6T SRAM cell has been the default choice for on-chip memory structures that are within the processor core such as register files and caches. DRAMs have been used for larger off-chip caches or main memories. An issue associated with DRAMs is that data is stored temporarily and thus require periodic refresh if the data needs to be held for extended time periods. On the other hand, a large fraction of the data consumed in the processor core is transient, especially data held in register files. In some sense, dynamic memory is actually a better choice for on-chip memory structures that provide temporary data storage, because the temporal characteristics of the data in the processor may reduce or even eliminate the need for refresh.

Since most on-chip memories require fast access to guarantee performance, 6T SRAM cells are generally used because they can be fast. However, recent circuit innovations like the 3T1D (3-transistor 1-diode) DRAM cell [14] demonstrate the possibility of building DRAMs, whose speed can match that of 6T-based SRAMs. While such a DRAM holds data temporarily and comparable access speeds can only be observed for a short interval of time after a write (or refresh), if the architecture can accommodate this fact it may be possible to replace on-chip SRAMs with DRAM structures. As shown throughout this paper, replacing SRAMs with DRAMs offers several advantages, which can lead to novel variation-tolerant memory architectures that can better track future process technology advances into the nanoscale regime.

A. Limitations of 6T SRAM Under Process Variations

In this section, we highlight several problems the traditional 6T SRAM faces as process variations get worse. Issues related to speed, stability, and area prevent the SRAM from easily scaling to nanoscale technologies.

1) Speed: Process variations can greatly affect the access time of an SRAM array. A typical 6T SRAM cell is illustrated in Figure 1a and the read and write operations are shown in Figure 1b. In order to read the stored data, one side of the cell typically discharges one of two precharged bitlines. Any variations in the gate length or threshold voltage in the cell's transistors (*T1* or *T2*) can vary the current driving capability of the read path. Variation also impacts the speed of the write path. As shown in previous works [4], [5], [13], the speed of SRAM array is very susceptible to process variations. In other words, one slow cell in the SRAM array can affect the speed of the entire structure. Since the slowest component determines the chip frequency, the degraded speed of SRAM will translate directly to performance loss seen for the entire microprocessor.

2) Stability: 6T SRAM cell stability will be a major hurdle for future VLSI designs due to increasing device-to-device mismatch. For reliable read operations, transistors *T1* and *T2* in Figure 1b must be carefully sized in order to prevent *nodeA* from rising and inadvertently flipping the bit. For writes, the combined effective resistance of the access transistor *T1* and write transistor *T7* must be low enough to overcome the pull-up strength of *T3* and flip the bit. Clearly, the read and write operations require a delicate balance of device sizes and transistor drive strengths to ensure proper operation. Unfortunately, increasing device mismatch leads to imbalances in the 6T cell that can compromise its ability to read or write reliably.

Our circuit-level 6T cell stability simulations reveal bit-level flip-rates on the order of 0.4% at the 32nm technology node, similar to [2]. Although this rate appears small, it may have large ramifications depending on available solutions. For example, in a data cache structure, line-based redundancy is straightforward to implement, but is ineffective because

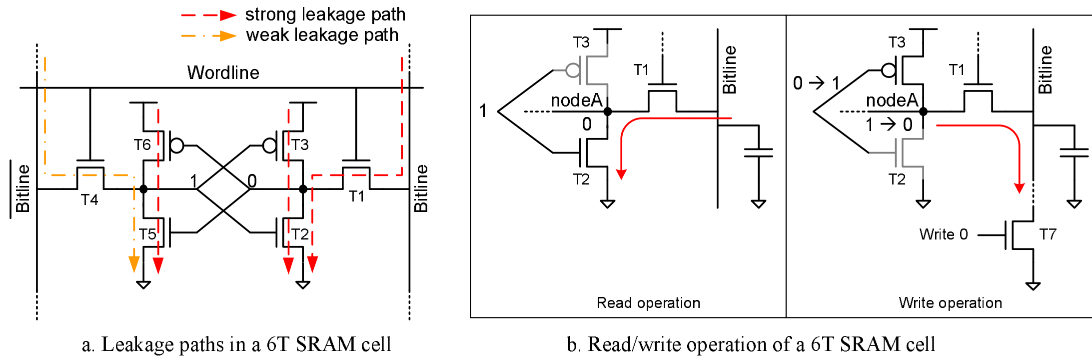


Fig. 1. Traditional 6T SRAM cell design

256-bit lines would experience a 64% probability of line failure (i.e., 1-0.996^{256}). Word-level (64-bit) granularity would experience a 23% failure rate but requires a much more complex implementation.

3) Power: Leakage power is another major concern for SRAM. It is well known that leakage power is poised to become the dominant source of power consumption in future digital chips. As shown in Figure 1a, there are three strong-leakage paths in one 6T SRAM cell since there is only one "off" transistor along the path. There is an additional weak-leakage path because there are stacked transistors along its path. Such an SRAM structure consumes considerable amounts of static current simply to preserve data in the SRAM. This may not be the most efficient use of power especially in the situation where a large portion of the data in the processor is only needed for a very short period of time. To make matters worse, process variation magnifies the leakage problem. Since there is an exponential relation between leakage current and threshold voltage, increasing variations in threshold voltage can cause more than 20 times leakage power variation across chips [18].

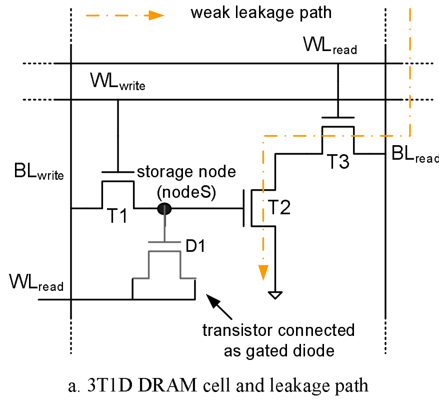
4) Area: Limited on-chip area and large memory requirement of modern systems call for very dense on-chip memory designs. In a 6T cell, most of the cell area is consumed to satisfy the design rule requirement for cross-coupled wire routing, contacts, and the well isolation between PMOS and NMOS transistors. To make a compact SRAM design, most of the transistors in the SRAM cell are close to minimum size. Unfortunately, process variation will have greatest impact on minimum size devices since the effects of gate length and threshold variations are more severe for smaller devices. To mitigate this impact, transistors in a 6T cell may be sized up. As will be shown in Section 4.1, the cell area must be doubled to achieve satisfactory yield. This approach is very inefficient. For a fixed area, such reduction of resource capacity would lead to significant performance loss in a microprocessor. Increasing cell size to overcome the detrimental effects of process variation eventually counteracts the area density benefit of technology scaling.

In a word, process variation will negatively impact the speed, stability, and power of traditional SRAM designs. Solutions like simply sizing up devices have unacceptable area and power penalties. A desire to continue the scaling trend for on-chip memories in nanoscale technologies drives us to seek out revolutionary memory circuit and architecture solutions.

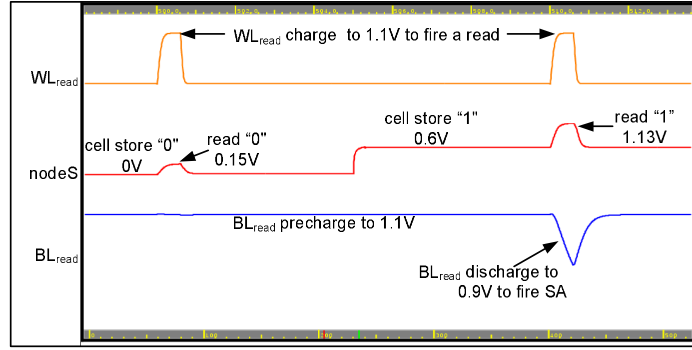
B. Introduction to Novel 3T1D DRAM

Recent circuit innovations in memory design provide an alternative way. Luk et al. proposed a novel 3T1D DRAM cell which offers speed comparable to a 6T SRAM cell for a limited period of time after writing the data [14], [16]. Chips fabricated in IBM's 130nm and 90nm processes demonstrate high-speed dynamic 3T1D memories. Based on this demonstration, we propose a novel memory architecture, which can potentially solve all of the major problems associated with SRAMs without much added circuit or architectural complexity. This section compares the speed, power, and area of the novel 3T1D DRAM to the traditional SRAM to demonstrate how the 3T1D DRAMs can be a suitable replacement for future on-chip memory designs.

1) Speed: Generally speaking, SRAMs are believed to be faster than DRAMs. DRAMs are traditionally comprised of 1T1C (1-transistor, 1-capacitor) cells with emphasis placed on density at the expense of speed. Furthermore, the destructive read of a 1T1C cell requires a writeback that immediately follows each read access. A 3T1C (3-transistor, 1-capacitor) DRAM cell does not suffer from destructive reads, obviating data writeback after each read access. However, the speed of a 3T1C cell is slower than a 6T SRAM cell. In comparison, the novel 3T1D (3-transistor, 1-diode) DRAM cell replaces the capacitor with a gated diode to solve this speed problem as shown in Figure 2a. This diode can be thought of as being a voltage-controlled capacitor with higher capacitance when storing a "1" and a lower capacitance when storing a "0." Each time the cell is read, the bottom side of this capacitor is also raised to VDD. Hspice simulation results, shown in Figure 2b, illustrate the operation of the 3T1D cell. By exploring the "amplification effect" of the diode [15], the voltage on the storage node is boosted by about 1.5-2.5 times the originally stored value if a "1" is stored when being read. Although the voltage on the storage node is only about 0.6V (degraded value), it is boosted to 1.13V when reading. This boosting strongly turns on the pull-down transistor (T2) and



a. 3T1D DRAM cell and leakage path



b. Timing of read operation in 3T1D cell

Fig. 2. Novel 3T1D DRAM cell design

rapidly discharges the bitline. Conversely, when a "0" is stored, there is little boosting (0.15V), which will keep the pull down transistor off during a read. As a result, the access speed can match the speed of 6T SRAM cells.

Although the speed of a 3T1D cell can be fast, this high-speed access is only valid for a small time period after each write to the cell. This is because the charge on the storage node leaks away over time. Figure 3 shows the relationship between cell access time versus the elapsed time passed after a write operation. With this stored charge leaking away, the amplification effect of the diode becomes smaller and the access time increases until finally it falls below the access speed of the 6T SRAM cell. In this paper, we re-define the *retention time* of a 3T1D memory as the time period during which the access speed can match that of a 6T SRAM cell with the same area (in the figure, about $5.8\mu s$ for nominal cells). Within this retention time, the memory can be accessed at the chip frequency. After this retention time passes, the memory cell has to be refreshed (re-written), otherwise that cell is considered invalid. Although the nominal retention time in our circuit simulations is very short (100s of nanoseconds to several microseconds), it is actually "long" enough for the temporary data-storage purposes of the processor, especially for register files.

Process variation can also negatively impact the 3T1D memory. For example, if the driving capability of the access transistors is reduced due to the variation of gate length or threshold voltage, the access time to the cell increases. This effect can otherwise be viewed as decreasing retention time, which is shown in Figure 3. Weaker than designed access transistors have the effect of shifting the access time curve to the left and cell retention time reduces (down to $4\mu s$). On the other hand, with stronger devices, retention time can increase. It is important to note that process variations do not necessarily impact operating frequency, which is the case for 6T SRAMs. In a 3T1D DRAM, process variations cause variations in retention times while still able to maintain the same nominal access speed. Moreover, variations in the speed of the decoder, sense amplifier, and other peripheral circuitry in the memory can be absorbed into the retention

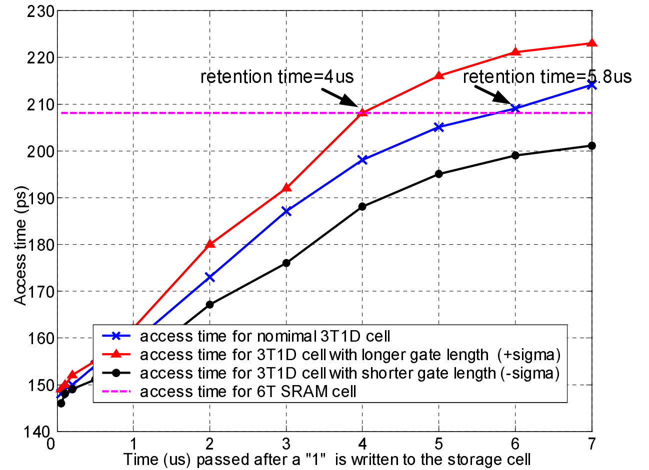


Fig. 3. Access time with retention time in 3T1D cell under process variation

time variation parameter. While write access speeds are less critical than reads [27], process variations in the write circuitry can also be absorbed into the retention time. This is because weaker circuitry associated with the write path only affect the charge delivered to the storage node during a write operation. Thus, the impact of process variations on a 3T1D cell can all be lumped into a single variable – the retention time of the cell. As detailed in Section V, modern out-of-order microprocessors can actually tolerate very large retention time variations and, thus, effectively eliminate the impact of device process variations in 3T1D-based memories.

2) Stability: The 3T1D DRAM cell does not suffer the cell stability issues previously seen in 6T SRAM cells, because there is no inherent fighting. Read operation occurs by simply discharging or charging the bitline, and write operation occurs by charging or discharging a dynamic storage node within the 3T1D cell. Except for the finite data retention time, a 3T1D DRAM cell is inherently stable.

3) Power: The 3T1D DRAM cell does not suffer the multitude of strong leakage paths previously seen in 6T SRAM cells. Hence, leakage power associated with 3T1D-based memories can be much smaller. If there is a "0" stored

Technology node	Min size. cell area for RF	Wire width	Wire thickness	Oxide thickness	Chip frequency
$65nm$	$2.56um^2$	$0.10um$	$0.20um$	$1.2nm$	$3.0GHz$
$45nm$	$1.26um^2$	$0.07um$	$0.14um$	$1.1nm$	$3.5GHz$
$32nm$	$0.62um^2$	$0.05um$	$0.10um$	$1.0nm$	$4.3GHz$

TABLE I
PARAMETERS FOR CIRCUIT SIMULATION

in the cell, there is only one weak leakage path given two stacked off transistors, shown in Figure 2a. If there is a "1" stored, since most of the time the stored value is a degraded value (only boosted high for a short period during read), there is only one slightly strong leakage path. Furthermore, if the stored charge has leaked away, the slightly strong leaking path becomes weak. In comparison, a 3T1D DRAM memory array has more than an order of magnitude lower leakage than a 6T SRAM memory array. The smaller number of leakage paths leads to lower nominal leakage and less leakage variability.

The structure of the 3T1D cell also benefits from having lower dynamic power during a write. For a 6T memory, one bitline must always be charged up to full rail (while the other is fully discharged) to successfully write the bit by overpowering previously stored data. In comparison, a 3T1D cell relies on a single-ended write, which only requires one bitline to be charged or discharged.

While a 3T1D cell saves dynamic power for writes, there is additional power during reads. The power overhead comes from the diode. If there is a memory read, the source voltage of the diode is raised, which consumes additional dynamic power. Also, if data is required beyond the retention time, a refresh is necessary incurring additional power. It is important to note, however, that this refresh may only occur on an as-needed basis.

4) Area: A 3T1D cell is much more area efficient compared to 6T SRAM cells, because the wire connection in a 3T1D cell is much simpler and there are no PMOS devices. This means the 3T1D cell can be smaller or, for the same area, the devices in a 3T1D cell can be larger to mitigate process variation. We emphasis here that most of the variation tolerance advantage of a 3T1D memory is not from the sized up devices, but by the ability to absorb retention time variation in the microarchitecture.

IV. EXPERIMENTAL METHODOLOGY

All the delay and power data presented in this paper are derived from Hspice circuit simulations. The minimum-size SRAM cell was designed in a commercial 130nm technology design kit. This design was layed out, simulated, and optimized for area, delay and power, to be comparable to commercial designs. We also designed and layed out the DRAM-based RF cell with the same area as the minimum-size SRAM cell. For comparison purposes, we also designed larger SRAM cells in which each transistor is double-sized. This results in 1.5× larger array areas for register files. Figure 4 plots a 4rd/2wr DRAM cell used in our RF simulation. Based on

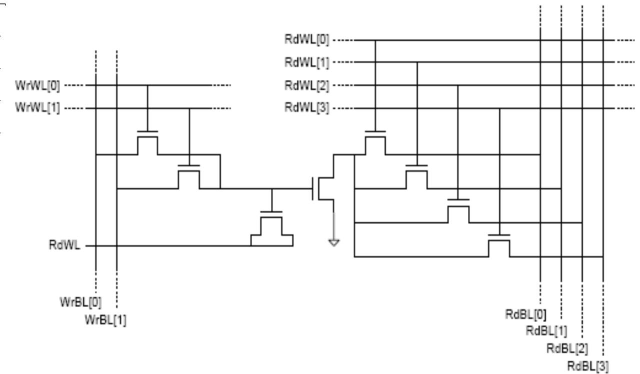


Fig. 4. DRAM-based RF cell with 4 read ports and 2 write ports

these original designs, we then scaled the designs to 65nm, 45nm, and 32nm technology nodes. All Hspice simulations for the three technology nodes are based on the newest version of the Predictive Technology Models (PTM) [28]. All wires were scaled with respect to technology and cell area. We assumed copper wires and use distributed-π models for wire delay simulation. Table I shows all the detailed circuit parameters of our design and all simulations were run at 1.1V and 80°C. This paper investigates a multiported register file.

We rely on Monte-Carlo simulation, similar to approaches found in [1]. This method considers both die-to-die and within-die variations and also handles correlations related to layout geometries. Recent experimental results verify that this method is very accurate; it has an error of 5% [6] which is sufficient for our architectural study. Given the absence of experimentally verified data in the 65nm technology node and beyond, we consider two situations for variation. The *typical* variation assumes $\sigma L/L_{nominal} = 5\%$ for within-die gate-length variations and $\sigma V_{th}/V_{th_{nominal}} = 10\%$ for threshold voltage variations. The *extreme* variation assumes $\sigma L/L_{nominal} = 7\%$ for within-die gate-length variations and $\sigma V_{th}/V_{th_{nominal}} = 15\%$ for threshold voltage variations. For both situations, we assume $\sigma L/L_{nominal} = 5\%$ for die-to-die gate length variation. These assumptions are comparable to the data forecast in [4]. For each Monte-Carlo simulation, 1000 versions of process variations are generated and simulated using the above parameters.

For architecture simulation, we assume a baseline machine with parameters listed in Table II which is comparable to the Alpha 21264 and POWER4. For our IPC simulations, we utilize the *sim-alpha* simulator [7].

We run an exhaustive set of architecture simulations to investigate the system-level impact of process variations. For example, each data point of Monte-Carlo simulation requires a corresponding architecture simulation in order to quantify the system-level impact of the variations. To manage this large number of simulations, we use 8 of the 26 SPEC2000 benchmarks and rely on Sim-Point for sampling [24]. Phansalkar et al. show that these 8 benchmarks (*crafty, applu, fma3d, gcc, gzip, mcf, mesa, twolf*) can adequately represent the entire

Configuration	Parameter
Issue Width	4 instructions
Issue Queues	20-entry INT,15-entry FP
Load Queue	32-entries
Store Queue	32-entries
Reorder Buffer	80-entry
Instruction Cache	64KB, 2-way Set Associative
Instruction TLB	128-entry Fully-Associative
Data TLB	128-entry Fully-Associative
Integer Functional Units	4 FUs
Floating Point Functional Units	2 FUs
L2 Cache	2MB 4-way
Branch Predictor	21264 Tournament Predictor

TABLE II

BASELINE PROCESSOR CONFIGURATION.

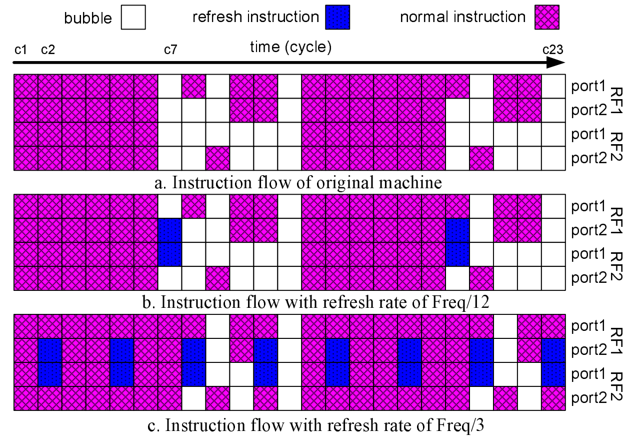


Fig. 5. Instructions flowing through register file in 23 clock cycles

SPEC2000 benchmark suite [21]. For each benchmark, 100 million instructions are simulated after fast forwarding to specific checkpoints. When we report single number results (performance or power) in this paper, they represent the harmonic mean of all simulated benchmarks. In addition to the mean, we also present results for worst-case benchmarks when appropriate.

V. PROCESS VARIATION TOLERANT REGISTER FILES

Based on our earlier discussion of DRAM cells and comparison to SRAM cells, this section investigates the architectural support needed to allow dynamic cells to replace traditional SRAM cells. 3T1D cells present many opportunities, but present challenges that must be overcome with system-level support. The major issue in 3T1D memory is its limited retention time and variations in this retention time. Given the relationship between access latency and retention time, the cell can only hold the data for a short time period to ensure the fast-access requirement of on-chip memories. After that time, the cell need to be refreshed or the data is lost. Refresh operations require a read and a subsequent writeback to the memory. A rudimentary refresh mechanism would add an extra read/write port specifically dedicated for refresh. However, this approach suffers considerable area and power overhead. Furthermore, refreshes may be infrequent and the extra port will likely be idle for long time periods. Instead, we opt for less costly refresh mechanisms that leverage existing ports in the memory used for normal access. Whenever a refresh is needed, a port is blocked from normal operation and used to refresh the data. This approach introduces a performance penalty because the refresh operation competes with normal instructions. For very short retention times, the penalty for this approach can be high. Moreover, cell-to-cell variations in retention times required by each 3T1D memory cells complicate this refresh strategy. Fortunately, under-utilization of processor resources in out-of-order machines provide ways to hide refresh operations, and the associated power and performance penalties can be low. Furthermore, given that most of the data flying in the processor core is transient, refresh operations are sometimes unnecessary, because memory entries are quickly overwritten or invalidated. Consequently, we observe negligible performance impact due to retention-time variation. This is very promising since the physical device variation can be transformed to retention time variation (using DRAM) and thus be absorbed by the dynamic behavior of OoO processor architecture.

A. Global Refresh Scheme for Register Files

The simplest way to refresh the register file is to use a global scheme. For every N clock cycles, one entry of the register file is refreshed. The refresh rate is $Freq/N$, where $Freq$ is the chip's operating frequency. The value of N depends on the retention time of the 3T1D memory cells. Longer retention times need larger N and, thus, less frequent memory refreshes. The minimum value of N is "1" which means that one read and one write port of the register file will be completely reserved for refresh operations. The minimum allowed retention time for a register file depends on the number of entries. For example, in a 80-entry register file, each entry must be able to sustain data for at least 80 clock cycles. For a 4.3GHz machine designed at the 32nm node, 80 cycles are equivalent to 18.6ns and we find that this minimum time can usually be met by the 3T1D cells we have simulated under typical variations.

The refresh operation itself can be quite simple. One option would be to utilize the existing instruction flow pipeline and use a special refresh instruction similar to a "move." This instruction is inserted into the register file read stage to read out the data entry. It then passes through the execution stage and writes the data back to the register file. It is important to check for pipeline hazards and not squash this instruction during a pipeline flush.

Given the under-utilization in modern superscalar processors, there are numerous bubbles in the pipeline, and consequently the refresh of the register file has a minimal impact on performance. For example, Figure 5a shows a snapshot of instructions flowing through the integer register file for 23 cycles in our simulated machine running "gzip" (a program

with heavy integer register file utilization). Figure 5b shows the same set of instructions with a refresh rate $Freq/12$ ($N = 12$). In this example, we find that none of the refresh operations conflict with normal instructions resulting in no loss of performance. In Figure 5c, given a 4X larger refresh rate (i.e. 4X smaller retention time) of $Freq/3$ ($N = 3$), some conflicts do occur and regular instructions are postponed. However, in this example with much tighter retention time requirements, existing instruction dependency slack hides these stalls and again there is no overall performance impact. This interesting outcome shows two important conclusions: 1) Many of the refresh operations can be absorbed into the existing pipeline (e.g. pipeline bubbles and dependency slack) resulting in small performance penalty. 2) Although device variation causes retention time variation, when incorporated into the processor architecture, it only introduces small performance variation.

The hardware implementation is relatively simple. Only two global counters are needed, as shown in Figure 6a. The clock counter counts N cycles to generate a refresh cycle pulse. The address counter counts up to the number of entries in the register file to decide which entry needs refreshing during that pulse. During the refresh cycle, a "block" signal is generated and passed back to the issue stage to block one pipe in the machine for one cycle. Meanwhile, a special refresh instruction is inserted into the pipeline and the entry will be refreshed 2 cycles later. Any dependent instruction during the 2 cycles can issue normally since data will be forwarded. A preceding instruction that writes to the same entry can proceed as normal since its results will be forwarded to the refresh instruction and the old value will not overwrite the new value. Because there are only two global counters, each with a small number of bits, the power and area overheads are negligible. Most of the power overhead comes from the refresh of the register file and the data propagated through two pipeline latches. We show power results in the next subsection.

B. Simulation Results of Global Refresh Scheme

This section presents the performance and power simulation results for the global refresh scheme described above. All forthcoming plots of simulation results are at the 32nm technology node where the most severe process variations occur. We also tabulate simulation results for other technology nodes so that the scaling trends can be observed. In these plots, the golden 6T SRAMs cells assume a minimum-size cell with ideal transistors (no process variations), 1X 6T and 2X 6T SRAM cells refer to minimum size and enlarged cells that both experience process variation, and 3T1D refers to a 3T1D DRAM cell that also experiences process variations and that has the same area as the golden 6T SRAM cell. All power, performance, and frequency results are normalized with respect to the golden 6T SRAM cell.

Figure 7a presents the frequency distribution, performance, and dynamic power of 6T SRAM-based register files assuming typical process variations. The top subplot presents the distribution of achievable register file clock frequencies in the presence of process variations. The middle subplot

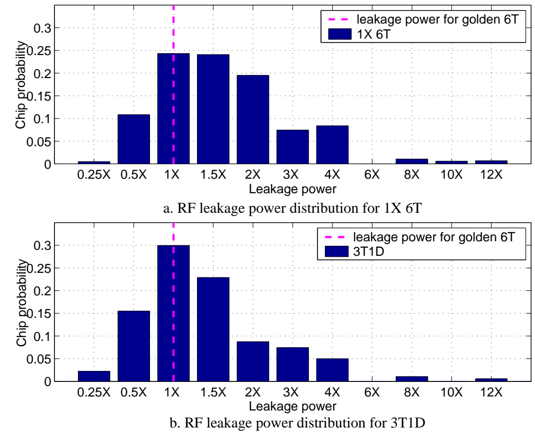


Fig. 8. Register file leakage power distribution under process variation

presents the maximum achievable chip performance in light of limited register file clock frequencies. A performance of 1, corresponding to the golden 6T is presented. The bottom subplot presents the resulting dynamic power for the register file across different clock frequencies. For the 1X 6T case (left half of Figure 7a), 99% of chips can only achieve normalized performance between 0.775-0.875, which corresponds to about a 20% performance loss attributable to process variations. When doubling cell size (2X 6T) to improve performance, 99% of chips can achieve normalized performance greater than 1 (right half of Figure 7a). However, this solution comes at the expense of at least a 1.5X increase in dynamic power and a 1.5X increase in area.

In contrast, Figure 7b presents similar plots for 3T1D register files. In this design, we hold clock frequency constant at the nominal rate that can be achieved by the golden 6T cell. The top subplot presents a distribution of required refresh rates of the register file (set by minimum retention time requirements). While a fairly broad distribution of refresh rates is observed among all chips, the performance doesn't vary too much. As demonstrated in the middle subplot, a normalized performance (averaged across simulated benchmarks) of 0.99 can be achieved by 99% of chips. Appreciable performance degradation is only observed when refresh rates increases above $Freq/2$. For completeness, we also plot the worst case benchmarks (gzip) and see little difference from the mean for 99% of the chips. Given the dependency slack and underutilized resources in the microarchitecture, very little performance loss is observed even with a broad range of retention time variations. The bottom subplot also shows that savings in dynamic power are possible. For 99% of the chips, at least 20% power savings is observed due to lower write power for the 3T1D. This dynamic power calculation includes the overhead power required for refresh which dominates when retention times are very low.

In addition to maintaining performance while dissipating less dynamic power, 3T1D memories inherently exhibit better leakage characteristics for reasons explained in Section III-

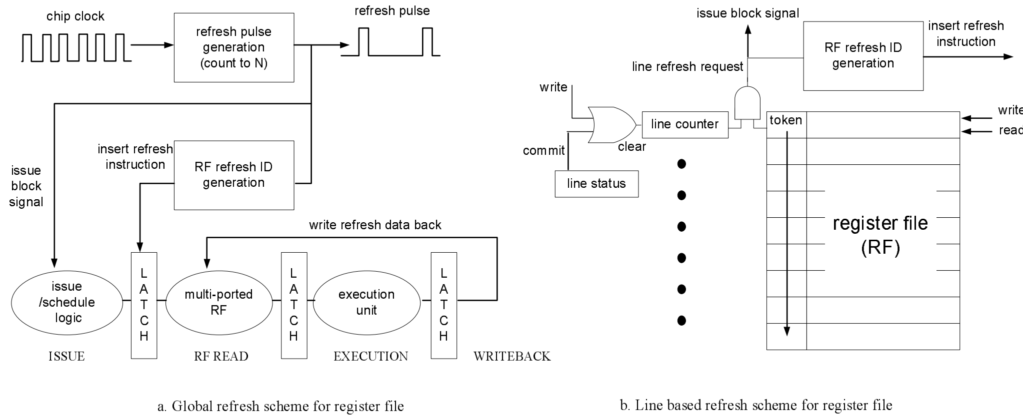


Fig. 6. Refresh scheme for register file

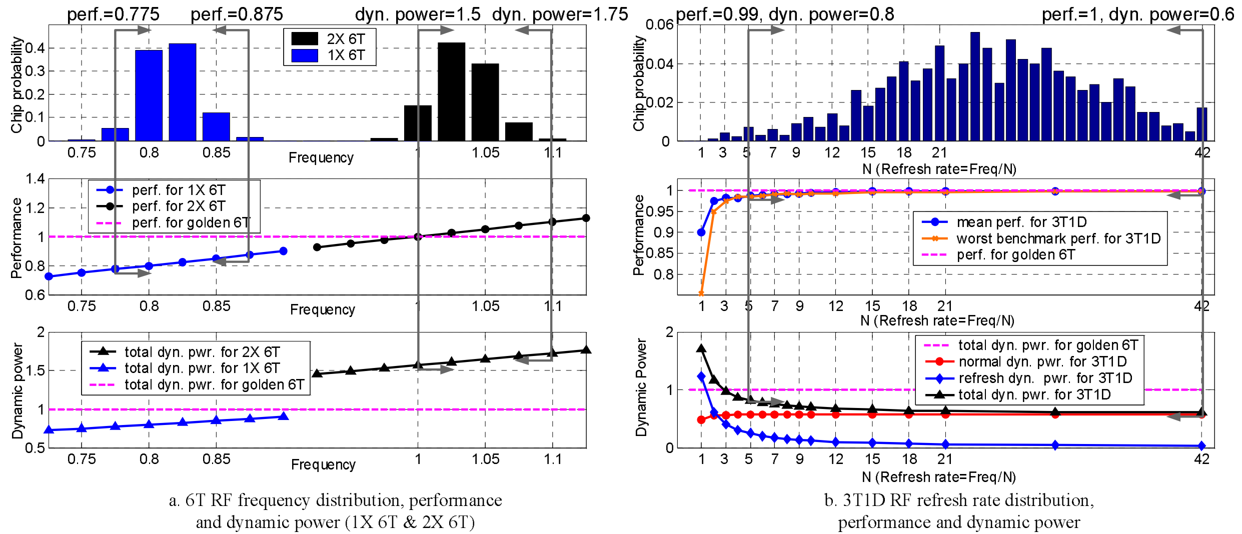


Fig. 7. Distribution, performance and power for 6T and 3T1D register files

B.3. Figure 8 plots the distribution of leakage power for 1X 6T and 3T1D register files as a result of process variations. For the 1X 6T-based design (Figure 8a), approximately 36% of chips exhibit leakage power equal to or less than that of a golden 6T design, while 37% of chips exhibit greater than 2X leakage power. In contrast, as shown in Figure 8b, 47% of 3T1D designs have leakage equal to or less than that of a golden 6T design. Only 18% of chips exhibit greater than 2X leakage power.

Table III provides a detailed summary of simulation data for three technology generations. Comparing the performance of 1X 6T and golden 6T SRAMS, roughly one generation of performance is lost due to process variation. On the other hand, 3T1D designs can maintain performance scaling trends while also achieving dynamic and leakage power advantages. Although retention time decreases with rising leakage in more advanced technologies, our simulations suggest significant headroom is available to accomodate this trend. Furthermore, we can accommodate lower retention times by by using more ports in the register file for refresh operations. In addition, process innovations such as high-K materials and FinFETs will likely curb future leakage increases.

C. *Line-Based Refresh Scheme for Register Files*

Although the global scheme provides good results, a fine-grained line-based refresh scheme can rescue chips that must otherwise be discarded. Because of on-chip process variations, each line in the register file can have different retention times. The previously described global refresh scheme is constrained by the worst-case line in a register file. If that worst-case line fails to meet the minimum retention time, the chip must be discarded. However, this is overly conservative since a majority of the chip may still be operable. Line-based schemes can address this problem by monitoring the retention time requirements of each line individually. Each line can have an associated counter to set the individual retention time and assert refresh requests. By tracking retention times on a per-line basis, we can also disable registers that have substantially short retention times. This can be accomplished by removing the register ID from the free register pool. This effectively

Tech. node	golden 6T, no variation Access time	Perf. (BIPS)	Mean Dyn. Pwr	Full Dyn. Pwr	Leakage Power	1X sized 6T, typical variation, median chip Access time	Perf. (BIPS)	Mean Dyn. Pwr	Full Dyn. Pwr	Leakage Power	3T1D, typical variation, median chip Retention time	Perf. (BIPS)	Mean Dyn. Pwr	Full Dyn. Pwr	Leakage Power
$65nm$	$243ps$	2.91	$6.01mw$	$20.89mw$	$0.59mw$	$296ps$	2.42	$5.20mw$	$18.07mw$	$0.59mw$	$918ns$	2.90	$3.46mw$	$13.76mw$	$0.52mw$
$45nm$	$212ps$	3.39	$4.87mw$	$16.99mw$	$1.68mw$	$265ps$	2.94	$4.21mw$	$14.69mw$	$1.68mw$	$664ns$	3.37	$3.03mw$	$12.11mw$	$0.98mw$
$32nm$	$188ps$	4.17	$4.34mw$	$15.03mw$	$4.98mw$	$237ps$	3.41	$3.56mw$	$12.33mw$	$4.98mw$	$513ns$	4.15	$2.76mw$	$10.90mw$	$4.47mw$

TABLE III

DETAILED SIMULATION RESULTS OF REGISTER FILES ACROSS THREE TECHNOLOGY NODES. RESULTS IN THE TABLE ARE ONLY PRESENTED FOR THE CORE MEMORY ARRAY. FULL DYNAMIC POWER INCLUDES FULLY UTILIZED PORTS, MEAN DYNAMIC POWER INCLUDES CLOCK GATING STATISTICS FROM ARCHITECTURAL SIMULATIONS.

reduces the capacity of the register file, resulting in a small loss of performance, but it allows the chip to function.

Another important benefit for the line-based scheme results from the transient nature of data in the register file. Most of the data held in the physical register file only needs to only survive for a short time period after which that register will be committed and new data written in. Whenever new data is written, the register is considered refreshed and the counter is reset. Our simulations shows that register files are written frequently and almost all registers are rewritten within their retention times. This characteristic offers a distinct advantage over the global scheme as there are no unnecessary refreshes. In the line-based scheme, lines are only refreshed if data must be held for extended time periods. This happens very infrequently in machines with physical register renaming renaming due to the commit and register reuse behavior of these architectures. Hence, the proposed line-based scheme greatly reduces the rate of explicit refresh operations, which further improves performance and power.

The line-based refresh scheme can be implemented with little added hardware complexity, as shown in Figure 6b. A counter added to each line can keep track of the retention time while being clocked at a fraction of the chip frequency. The counter starts when the register is renamed and valid data is written and resets once the register is committed. The counter control can leverage existing logic that specifies the state of each register. In order to guarantee correct operation under all conditions, a one-bit token circulates the register file to negotiate instances where multiple lines require refresh in the same cycle. A refresh operation only occurs if a line reaches its retention time while holding on to the refresh token, and the refresh operation can occur as described in the global scheme.

While area overhead (10%) associated with the line counters and token passing logic can not be ignored, this scheme significantly reduces refresh power. We find that the reduction in refresh power offsets the additional power overhead for line counters.

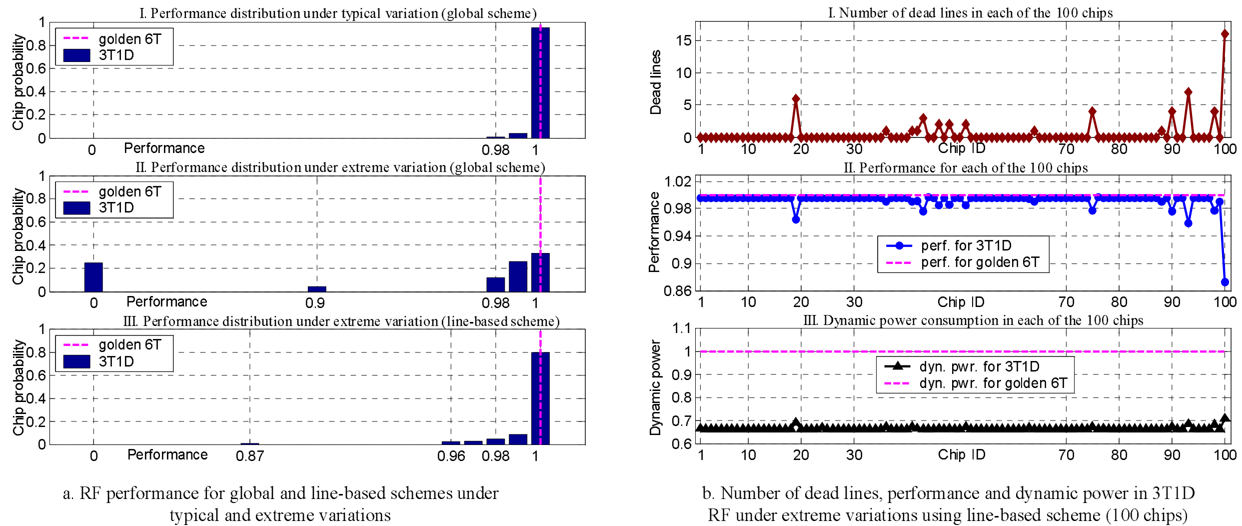
D. Simulation Results of Line-based Scheme

Figure 9a compares the performance of global and line-based schemes in register files under typical and extreme variations. With the global refresh scheme described earlier, extreme process variations lead to a significant number (20%) of discarded chips due to dead lines in the register file (middle subplot) and performance degrades relative to typical variations (top subplot). Because the line-based scheme can distinguish between good and bad lines, it works well even under extreme variations with very little performance loss. As seen in the bottom subplot, 97% of the chips can still have a relative performance larger than 0.97.

In order to show the benefit of the line-based scheme, we randomly pick 100 chips from our Monte-Carlo simulations of 1000 chips assuming extreme variations, and plot the number of dead lines, performance and dynamic power in Figure 9b. Several conclusion can be drawn from this figure. First, most of the chips tend to have the same performance and dynamic power using the line-based scheme (middle and bottom subplots). In contrast, the bottom subplot of Figure 7b shows that individual chips using the global scheme consume very different amounts of dynamic power according to their individual refresh rates. This can be explained by the very sparse occurrence of explicit refresh operations using the line-based scheme. Including all power overheads, almost all the 3T1D chips save about 34% dynamic power compared to the golden 6T design. The second conclusion we draw from this figure is that some chips can suffer a larger performance loss (e.g., chip 19 & chip 100). The middle subplot shows that the performance loss tracks the number of dead lines. Die-to-die variations can account for the large collection of dead lines for certain chips. Chip-level variability solutions such as adaptive body biasing [26] ought to reduce these die-to-die variations. Similarly, we see that some chips have larger than average power consumption (again, see chips 19 & 100 in the bottom subplot). This can be attributed to the fact that these chips have shorter average retention times and require more frequent refreshes. Again, this is caused by die-to-die variations and can be solved relatively easily by chip-level solutions.

VI. CONCLUSION

Microprocessors tolerant to extreme process variations in future nanoscale technologies will be at the forefront of



a. RF performance for global and line-based schemes under typical and extreme variations

b. Number of dead lines, performance and dynamic power in 3T1D RF under extreme variations using line-based scheme (100 chips)

Fig. 9. Register file using line-based scheme under extreme variations

innovation for years to come. This paper proposes novel process-variation tolerant on-chip memory architectures based on a 3T1D dynamic memory cell. The 3T1D DRAM cell is an attractive alternative to conventional SRAM cells for next-generation on-chip memory designs since they offer better tolerance to process variations that impact performance, cell stability, and leakage power.

By leveraging modern processor architecture and transient data characteristics, significant performance and power benefits have been demonstrated for register file designed in 3T1D memory. But this technique can also be used in other on chip memory dominant units such as issue queue, branch predictor and caches. These promising results suggest that dynamic memories can replace existing static memories as a comprehensive solution for process variations.

ACKNOWLEDGEMENTS

This work was supported by NSF grant CCF-0429782, the Fulbright program, the Generalitat de Catalunya (2005SGR00950), and the Ministerio de Educacion y Ciencia (TIN2004-07739-C02-01).

REFERENCES

[1] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. In *International Conference on Computer-Aided Design*, November 2003.

[2] A. Agarwal, B. C. Paul, H. Mahmoodi, A. Datta, and K. Roy. A process-tolerant cache architecture for improved yield in nanoscale technologies. *IEEE Transactions on Very Large Scale Integration Systems*, 13(1), January 2005.

[3] A. J. Bhavnagarwala, X. Tang, and J. D. Meindl. The impact of intrinsic device fluctuations on CMOS SRAM cell stability. *IEEE Journal of Solid-State Circuits*, 36(4), April 2001.

[4] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variation and impact on circuits and microarchitecture. In *40th Design Automation Conference*, June 2003.

[5] 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. *Journal of Solid-State Circuits*, 37(2), February 2002.

[6] B. Cline, K. Chopra, and D. Blaauw. Analysis and modeling of CD variation for statistical static timing. In *International Conference on Computer-Aided Design*, November 2006.

[7] R. Desikan, D. Burger, S. Keckler, and T. Austin. Sim-Alpha: a validated, execution-driven Alpha 21264 simulator. In *TR-01-23, CS Department, University of Texas*, 2001.

[8] J. Donald and M. Martonosi. Power efficiency for variation-tolerant multicore processors. In *International Symposium on Low Power Electronics and Design*, octoboer 2006.

[9] E. Humenay, D. Tarjan, W. Huang, and K. Skadron. Impact of parameter variations on multicore architectures. In *Workshop on Architectural Support for Gigascale Integration (ASGI-06, held in conjuction with ISCA-33)*, 2006.

[10] M. Khellah, Y. Ye, N. S. Kim, D. Somasekhar, G. Pandya, A. Farhang, K. Zhang, C. Webb, and V. De. Wordline and bitline pulsing schemes for improving SRAM cell stability in low-Vcc 65nm CMOS designs. In *2006 Sympsia on VLSI Technology and Circuits*, June 2006.

[11] N. S. Kim, T. Kgil, K. Bowman, V. De, and T. Mudge. Total power-optimal pipelining and parallel processing under process variations in nanometer technology. In *International Conference on Computer-Aided Design*, November 2005.

[12] X. Liang and D. Brooks. Microarchitecture parameter selection to optimize system performance under process variation. In *International Conference on Computer-Aided Design*, November 2006.

[13] X. Liang and D. Brooks. Mitigating the impact of process variations on processor register files and execution units. In *39th IEEE International Symposium on Microarchitecture*, December 2006.

[14] W. K. Luk, J. Cai, R. H. Dennard, M. J. Immediato, and S. V. Kosonocky. A 3-transistor DRAM cell with gated diode for enhanced speed and retention time. In *2006 Sympsia on VLSI Technology and Circuits*, June 2006.

[15] W. K. Luk and R. H. Dennard. Gated diode amplifiers. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 52(5), May 2005.

[16] W. K. Luk and R. H. Dennard. A novel dynamic memory cell with internal voltage gain. *Journal of Solid-State Circuits*, 40(4), April 2005.

[17] D. Marculescu and E. Talpes. Variability and energy awareness: A microarchitecture-level perspective. In *DAC-42*, June 2005.

[18] K. Meng and R. Joseph. Process variation aware cache leakage management. In *International Symposium on Low Power Electronics and Design*, octoboer 2006.

[19] S. Narendra, A. Keshavarzi, B. Bloechel, S. Borkar, and V. De. Forward body bias for microprocessors in 130-nm technology generation and beyond. In *IEEE Journal of Solid-State Circuits, Vol. 38, No. 5*, May 2003.

[20] S. Ozdemir, D. Sinha, G. Memik, J. Adams, and H. Zhou. Yield-aware cache architectures. In *39th IEEE International Symposium on Microarchitecture*, December 2006.

[21] A. Phansalkar, A. Joshi, L. Eeckhout, and L. K. John. Measuring program similarity: Experiments with SPEC CPU benchmark suites. In *IEEE International Symposium on Performance Analysis of Systems and Software*, March 2005.

[22] H. Pilo, J. Barwin, G. Braceras, C. Browning, S. Burns, J. Gabric, S. Lamphier, M. Miller, A. Roberts, and F. Towler. An SRAM design in 65nm and 45nm technology nodes featuring read and write-assist circuits to expand operating voltage. In *2006 Sympsia on VLSI Technology and Circuits*, June 2006.

[23] B. F. Romanescu, S. Ozev, and D. J. Sorin. Quantifying the impact of process variability on microprocessor behavior. In *2nd Workshop on Architectural Reliability*, 2006.

[24] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder. Automatically characterizing large scale program behavior. In *International Conference on Architectural Support for Programming Languages and Operating Systems*, October 2002.

[25] A. Tiwari, S. R. Sarangi, and J. Torrellas. Recycle: Pipeline adaptation to tolerate process variation. In *Proceedings of the International Symposium on Computer Architecture*, 2007.
[26] J. Tschanz, J. Kao, and S. Narendra. Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. In *Journal of Solid-State Circuits, Vol. 37, No. 11*, November 2002.
[27] S. Wijeratne et al. A 9GHz 65nm Intel Pentium 4 processor integer execution core. In *Proc. ISSCC*, Jan 2006.
[28] W. Zhao and Y. Cao. New generation of predictive technology model for sub-45nm design exploration. In *IEEE International Symposium on Quality Electronic Design*, 2006.