

NBTI Tolerant Microarchitecture Design in the Presence of Process Variation

Xin Fu, Tao Li and José Fortes

Department of Electrical and Computer Engineering, University of Florida, Gainesville, Florida, USA, 32611
xinfu@ufl.edu, taoli@ece.ufl.edu, fortes@acis.ufl.edu

Abstract—Negative bias temperature instability (NBTI), which reduces the lifetime of PMOS transistors, is becoming a growing reliability concern for sub-micrometer CMOS technologies. Parametric variation introduced by nano-scale device fabrication inaccuracy can exacerbate the PMOS transistor wear-out problem and further reduce the reliable lifetime of microprocessors. In this work, we propose microarchitecture design techniques to combat the combined effect of NBTI and process variation (PV) on the reliability of high-performance microprocessors. Experimental evaluation shows our proposed process variation aware (PV-aware) NBTI tolerant microarchitecture design techniques can considerably improve the lifetime of reliability operation while achieving an attractive trade-off with performance and power.

1 INTRODUCTION

Negative bias temperature instability (NBTI) is a considerable reliability concern for sub-micrometer CMOS technologies. NBTI occurs in PMOS devices when the gate-source voltage is negative ($V_{gs} = -V_{dd}$). NBTI increases the threshold voltage (V_{th}) and reduces the drive current (I_{dsat}), which causes degradation in circuit speed and requires a minimum voltage (V_{min}) increase in storage cells to keep the content. Eventually, this will lead to failures in logic circuits and storage structures due to timing violations or V_{min} limitations. The NBTI effect in PMOS transistors, which stems from an electro-mechanical reaction involving the electric field, holes, Si-H bonds, and temperature, is not a recently discovered wear-out mechanism. It was originally observed in the early phases of CMOS development (almost 40 years ago), but was not considered important because of the low electric fields under normal operating conditions. However, technology scaling has resulted in the convergence of several factors (e.g. the introduction of nitrided oxides, the increase in gate oxide fields, and operating temperature), which have made NBTI the most critical reliability concern for deep sub-micrometer transistors [1, 2, 3]. For example, it has been observed that NBTI can increase V_{th} by as much as 50mV for devices operating at 1.2V or below [2] and the circuit performance degradation may extend up to 20% in 10 years [3]. Industry and academia have expressed interest in this research over the past few years, and attempting to understand, model, and characterize the effect of NBTI at the device level [4, 5, 6]. Circuit and architectural techniques for mitigating/tolerating NBTI have been proposed in [7, 8]. These studies, however, did not consider the impact of process variation.

As CMOS process technology is scaled down, process variation (PV), the divergence of transistor process parameters

from their nominal specifications, results in variability in circuit performance/power and has become a major challenge in the design and fabrication of future microprocessors [9, 10, 11, 12]. For example, chip frequency can be degraded by as much as 30% in 45nm process technology due to process variation [10] and a 20x increase in leakage power consumption is reported in [9]. PV is caused by the difficulty in controlling the sub-wavelength lithography and channel doping as process technology scales. Process variation consists of die-to-die (D2D) and within-die (WID) variations. Die-to-die variation consists of parameter fluctuations across dies and wafers, whereas within-die variation refers to variations of design parameters within a single die. As technology scales, within-die variation, which is the primary focus of this study, has become more significant and is a growing threat to future microprocessor design [9, 10]. The impact of PV on processor frequency and leakage power consumption has recently motivated several architecture and system level proposals for PV mitigation [13, 14].

The PMOS degradation (i.e. wear-out) problem due to NBTI is aggravated in the presence of process variation. Under the impact of PV, circuit operating frequency decreases significantly after the chip is fabricated (frequency is determined by the slowest critical path). The NBTI effect further exacerbates circuit performance degradation during chip operation due to increased V_{th} . Consequently, the decreasing circuit operating frequency is a cumulative effect of both PV and NBTI. Current PV-tolerant mechanisms largely ignore the NBTI wear-out problem. On the other hand, existing NBTI-tolerant techniques lack the ability to address the deleterious impact of PV. As a result, the chip can still suffer a significant frequency loss and increased power overhead even though the NBTI-tolerant mechanisms are applied. In the upcoming nano-/atom- scale transistor design era, microarchitecture design techniques which can effectively address the combined PV and NBTI effect are greatly needed.

In this paper, we show that simply combining PV mitigation techniques with NBTI recovery mechanisms cannot efficiently address the aggregated effect. Observing that process variation has both positive and negative effects on circuits, we take advantage of the positive effects in NBTI-tolerant design. We propose three microarchitecture NBTI reliability enhancements in the presence of process variation which mitigate the detrimental impact of PV and NBTI simultaneously, while achieving attractive trade-offs among chip performance, power, lifetime, and area overhead. We show that the proposed techniques can be applied to a wide range of microarchitecture structures, leading to significant reliability and performance improvements at the chip level. The contributions of this work are:

- We observe that microarchitecture designs that exploit the positive interplay between PV and NBTI can significantly improve the trade-offs among performance, reliability, and power. Unfortunately, a simple combination of PV mitigation techniques and NBTI recovery mechanisms lacks the capability of exploiting the opportunity to optimize their interaction.
- We propose techniques that can leverage the positive interplay between PV and NBTI while alleviating the negative interaction between the two. The proposed optimization 1 (O1) switches the read ports in multi-ported register files to migrate the NBTI effect to the ports under positive PV impact. This technique can also be extended to other multi-ported structures such as the issue queue. The proposed optimization 2 (O2) explores program narrow width values to mitigate the NBTI degradation in functional units. Meanwhile, it leverages the gained NBTI mitigation to balance the wear-out effect within the units under the negative impact of PV. The proposed optimization 3 (O3) applies an adaptive inversion scheme (a NBTI-tolerant mechanism) to different cache regions. The percentage of the cache line inverted within a cache region is determined by the impact of PV on that region. Using PV-aware cache line inversion allows us to minimize performance degradation while achieving desired chip lifetime. Experimental results show that at the chip level, the aggregated effect of our proposed optimizations improves the NBTI&PV_efficiency (a metric that describes the efficiency of addressing the NBTI and PV effect) by 117% compared to the baseline case without any optimization. In addition, our schemes outperform approaches that simply combine NBTI and PV mitigation techniques by 21%.

The rest of this paper is organized as follows. Section 2 provides background on NBTI and process variation and discusses the interaction between the two. Section 3 proposes PV-aware NBTI-tolerant microarchitecture designs. Section 4 presents our experimental methodology. In Section 5, we evaluate lifetime reliability enhancement, performance, and power impact of the proposed approaches. Section 6 presents related work and Section 7 concludes the paper.

2 BACKGROUND

In this section, we illustrate the effect of NBTI on PMOS transistors and describe mechanisms to recover NBTI degradation. Process variation and PV mitigation techniques are described in Section 2.2. The interaction between NBTI and PV is discussed in Section 2.3.

2.1 Negative Bias Temperature Instability (NBTI)

NBTI is the result of interface trap generation in the silicon/oxide interface of PMOS transistors. When the PMOS transistor is under negative voltage, the silicon-hydrogen bonds at the silicon/oxide interface can easily break and generate interface traps (N_{IT}). N_{IT} captures electrons flowing from the source to the drain and increases the PMOS threshold voltage. As a result, the transistor becomes slower and can cause failures when the delay exceeds timing specifications. NBTI leads to failures in the storage cell as well. Higher V_{th}

requires a higher V_{min} to keep the content and V_{min} in the cell may not be able to satisfy this requirement due to limited power budget. Note that PBTI (Positive Bias Temperature Instability) also occurs in NMOS transistors. However, its impact is negligible compared to the NBTI effect in PMOS transistors [15]. NBTI degradation can be recovered when the positive voltage is set at the gate of PMOS transistors. It helps to heal the interface traps generated, which partially recovers V_{th} . Thus, a PMOS experiences the period of either stress mode (gate is set as “0”) or recovery mode (gate is set as “1”) during its lifetime. The NBTI degradation is partially recovered once the stress is moved. Therefore, minimizing the period during which negative voltage is applied at the gate of PMOS can reduce the NBTI effect. Other methods, such as resizing PMOS or reducing the operating voltage, can also be applied to mitigate NBTI degradation [16, 17]. As discussed in [8], considering performance, power, and area overhead introduced, reducing the amount of time PMOS under stress outperforms other NBTI mitigation methods.

To mitigate NBTI degradation in combinational logic units, [8] proposed the use of special vectors as input into the units when they are idle; avoiding the aggressive stress on a specific PMOS. As a result, PMOS transistors in the units degrade evenly and their lifetime is extended since lifetime is determined by the most degraded PMOS. In storage cell (e.g. 6T SRAM) based structures (e.g. register file and cache), there is always one PMOS under stress and another under recovery. Therefore, the best NBTI degradation scenario is to degrade the two PMOS in the SRAM evenly. Storing “0” and “1” 50% of the time can achieve balanced NBTI degradation. To achieve this goal, [8] observed that on average, a register file entry is free (time between the release and the next write operation) around 50% of the time and proposed to invert the register file entry while in the free state. In addition, [8] proposed to invalidate and store the sampled inverted values into 50% of the L1 cache lines during the entire lifetime to statistically degrade the two PMOS in each SRAM bit evenly.

Guardbanding, as a conservative approach and a last resort, can be used to tolerate NBTI degradation. Guardbanding reduces the processor frequency or increases the minimal voltage to defend against the expected degradation in logic circuits or storage structures during the targeted microprocessor lifetime. For instance, in [18], 20% of the cycle time is reserved to combat NBTI degradation. Mitigating NBTI degradation can reduce the necessity of guardbanding, leading to improvements in frequency and power savings. However, NBTI mitigation techniques can cause performance penalties and power overhead, making it a poor choice if the overhead outweighs that of guardbanding.

$NBTI_{efficiency}$ (shown as Eq.1) is proposed in [8] to evaluate the efficiency of NBTI tolerant schemes. It quantifies the trade-off among performance (Delay), power and area overhead (TDP), and lifetime (the amount of required NBTIguardband). The Delay and TDP obtained by the technique will be normalized to the case without NBTI and PV effects. As can be seen, lower $NBTI_{efficiency}$ implies an improved approach and the optimum technique will achieve a $NBTI_{efficiency}$ of 1 since both the Delay and TDP will be 1, and the NBTIguardband is equal to zero.

$$NBTEfficiency = (Delay \cdot (1 + NBTEguardband))^3 \cdot TDP \quad (\text{Eq. 1})$$

2.2 Process Variation (PV)

Process variation is a combination of random effects (e.g. random dopant fluctuations) and systematic effects (e.g. lithographic lens aberrations) that occur during transistor manufacturing. Random variation refers to random fluctuations in parameters from die-to-die and device-to-device. Systematic variation, on the other hand, refers to the layout-dependant variation through which nearby devices share similar parameters. D2D variation primarily presents as a random variation, whereas WID variation is composed of both random and systematic variation.

A chip may experience considerable frequency loss or leakage power consumption due to the impact of PV. Variable-latency (VL) techniques have been proposed to compensate for frequency loss due to PV [13]. Take multi-ported register files (RF) as an example. For each read port in the register file, RF entries are partitioned into fast and slow entries based on the SRAM read delay. Read operations are assumed to complete in one cycle in fast entries, but take two cycles in slow entries. Slow entries are not accounted for when determining the operating frequency of the register file. $n\%$ VL-RF defines the RF frequency based on the slowest read time of the fastest $n\%$ RF entries for each read port. The frequency is pre-determined by testing the read ports in each RF entry. In VL-RF, it is possible that a RF entry will have both slow and fast read ports. When a slow port is assigned to a read operation, port switching (PS) is applied to switch from the slow port to a fast port in order to avoid the one cycle stall in the pipe belonging to the slow port. Note that stalls in the pipe reduce the issue bandwidth and, therefore, the IPC. [14] proposed applying fine-grained body biasing (FGBB) to mitigate the V_{th} variation within a single chip. The chip is partitioned into several sections, called cells. FGBB applies different body bias to each cell. Body bias (BB) is a voltage applied between the source or drain and substrate to adjust the V_{th} . Forward body biasing (FBB) decreases the V_{th} , decreasing the delay of the transistor, but makes it leakier. On contrary, reverse body biasing (RBB) increases the V_{th} , creating a less leaky, but slower, transistor.

2.3 The Interplay between NBTI and PV

As described earlier, both NBTI and PV affect PMOS V_{th} . Therefore, guardbanding should consider the potential V_{th} increase contributed by all factors. Only targeting on NBTI (or PV) underestimates the guardband requirement and results in a shorter lifetime. This is because the frequency loss and power overhead caused by PV (or NBTI) is not counted. On the other hand, simply adding a NBTI guardband to the PV guardband will overestimate the actual guardband investment since doing so conservatively assumes the worst case scenario and ignores the benign impact of PV on NBTI, which helps reduce the guardband. The excessive guardband causes unnecessary frequency loss and power overhead.

Since parameters vary around their nominal design specification, PV can have both positive and negative effects on transistor characteristics: it either decreases V_{th} ($-\Delta V_{th}$) or

increases V_{th} ($+\Delta V_{th}$). NBTI degradation only increases V_{th} , but the amount of increase on PMOS V_{th} varies significantly due to the different stress period. NBTI impact can be generally described as either high V_{th} increase ($high_ \Delta V_{th}$) or low V_{th} increase ($low_ \Delta V_{th}$). We can classify the aggregated effect of PV and NBTI into four categories: $-\Delta V_{th}$ & $high_ \Delta V_{th}$, $-\Delta V_{th}$ & $low_ \Delta V_{th}$, $+\Delta V_{th}$ & $high_ \Delta V_{th}$ and $+\Delta V_{th}$ & $low_ \Delta V_{th}$. The guardband will be as high as the sum of NBTI and PV guardbands if $+\Delta V_{th}$ & $high_ \Delta V_{th}$ dominates. Note that NBTI is a temporal effect, its impact on V_{th} dynamically changes across runtime during the lifetime, depending on the fraction of time its gate is set as “0”. The $high_ \Delta V_{th}$ shift can be compensated by PMOS with $-\Delta V_{th}$ with low performance penalty and power overhead. Therefore, the total guardband can be reduced to the $\max(-\Delta V_{th} \& high_ \Delta V_{th}, +\Delta V_{th} \& low_ \Delta V_{th})$ and a large amount of frequency and power savings is reclaimed. In an ideal scenario, where all positive effects of PV are exploited to mitigate the NBTI degradation, guardband will decrease to as low as the PV guardband. Figure 1 illustrates the difference between the conservatively estimated guardband with the optimized one which considers the interaction between NBTI and PV. The difference can be as large as 36% based on our evaluation.

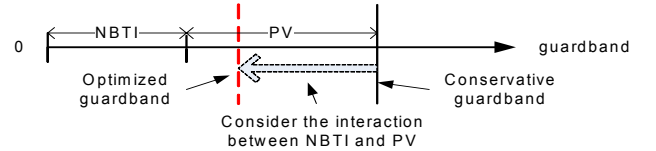


Figure 1. Different Guardband Settings for Tolerating NBTI

As discussed above, to achieve an optimized NBTI+PV guardband setting, it is important to consider the interaction between NBTI and PV. However, to our knowledge, existing NBTI and PV tolerant mechanisms [8, 13, 14] address the two factors individually and separately. In this paper, we propose several cost-effective PV-aware NBTI tolerant methodologies. To our knowledge, our work is the first attempt to consider NBTI and PV simultaneously while taking advantage of the positive interplay between the two to improve reliability efficiency.

3 PROCESS VARIATION AWARE NBTI TOLERANT MICROARCHITECTURE

In this Section, we argue that simply putting NBTI and PV tolerant techniques together can only reduce the total guardband requirement to a limited extent. Moreover, even though it can maximally reduce the guardband in some cases, it results in a large performance penalty. To efficiently reduce the total guardbanding while minimizing the negative impact on performance and power, we propose a set of PV-aware NBTI tolerant techniques for different types of microarchitecture structures that can exploit the positive interaction between NBTI and PV.

3.1 Motivation

In order to reduce the required NBTI and PV guardbands, one can apply NBTI tolerant and PV mitigation techniques together. This will mitigate the NBTI degradation and the deleterious PV effect independently. Take a multi-ported register file (RF) as an example. It is comprised of combinational logic circuits (decoders, wordlines, bitlines, and output amplifiers) and storage cells (SRAM based RF entries). The NBTI mitigation techniques that target logic circuits and storage cells introduced in Section 2.1 can be applied to reduce NBTI guardband. The NBTI guardband of the entire RF is determined by the highest NBTI guardband of the two parts. Meanwhile, the VL+PS (e.g. variable latency and port switching) scheme can be applied to the RF to reduce the frequency loss caused by PV and to minimize the PV guardband. However, as our evaluation results show in Section 5, simply putting the NBTI and PV mitigation techniques together only reduces the PV guardband and even has a negative effect on NBTI guardband because the PV mitigation technique exacerbates the NBTI degradation. The reason is that this method largely ignores the interplay between NBTI and PV and loses the opportunity to reduce the total guardband further. Since the ultimate goal of NBTI mitigation techniques is the same for different microarchitecture structures, one can expect that similar scenarios occur in other structures (e.g. issue queue, functional units). Figure 2 illustrates the limitation of the simple NBTI+PV mitigation technique.

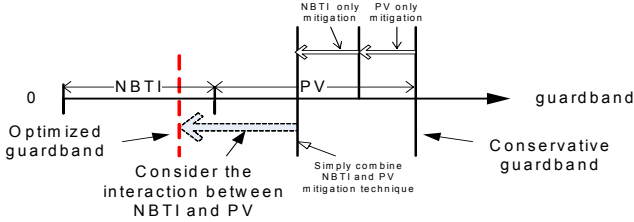


Figure 2. The Limitation of Simply Combining NBTI and PV Mitigation Techniques

Note that with a considerable performance and power overhead, it is still possible for the simple combined approach to reduce the total guardbands by a significant margin. However, as shown in Eq-1, guardband is not the only factor that determines the efficiency of the proposed techniques. The trade-off between reliability and performance/power should also be considered. The interaction between NBTI and PV provides the opportunity to minimize the performance penalty or power overhead without degrading the guardband enhancement obtained by the combined technique.

To summarize, simply combining NBTI with PV mitigation techniques lacks the capability to exploit the positive interaction between NBTI and PV which is beneficial to achieve either a lower guardband or less performance penalty and power overhead.

3.2 PV-aware NBTI Mitigation for Multi-ported based Microarchitecture Structures

In this Section, we present the proposed techniques in light of register file (RF) design since the RF is a representative multi-ported microarchitecture structure.

In a multi-ported RF, the RF delay is dominated by the read access time since write access time is not as delay critical as read access time [19]. In this study, we focus on RF read access and leave write access as future work. Figure 3 presents the 2-read port RF with detailed read port design. Only one bit cell is shown in this Figure due to space limitations. As it shows, a read port includes two wordline (the inverter) and two bitline transistors. The read access time consists of the wordline charge delay and the bitline discharge delay. Variation of the four transistors will cause a difference in the read access time of each read port. It will further affect the RF frequency, which is determined by the slowest read access time. Therefore, the effect of PV and NBTI on the read port should be accounted for by guardband estimation.

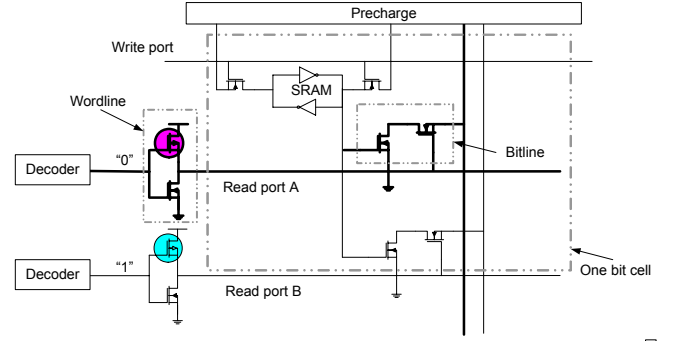


Figure 3. 2-Read Port Register Files with Detailed Read Port Design

When a read port is selected to perform the read operation (e.g. read port A in Figure 3), the decoder will trigger the wordline associated with that port. This causes a negative voltage to be set at the PMOS gate in the inverter and triggers the NBTI degradation. On the other hand, if the port is not selected (e.g. port B in Figure 3), the positive voltage is set at the PMOS gate, putting that PMOS under the recovery mode. As can be seen, the port is under stress mode whenever it is enabled for read operation. Therefore, reducing the port utilization can help mitigate NBTI degradation.

Based on the above observation, we propose microarchitecture optimization 1 (O1) which assigns higher utilization to the ports with shorter read access times. By doing so, the ports with longer read access times suffer much less NBTI degradation since their utilization decreases. As can be seen, O1 leverages the interaction between NBTI and PV by migrating more NBTI degradation to the ports with low V_{th} (due to PV). Therefore, it minimizes the case of $+\Delta V_{th}$ & $high_ \Delta V_{th}$ and efficiently reduces the NBTI guardband requirement. Since VL has been proved as an efficient PV mitigation method [13], we use VL technique in O1 to reduce the PV guardband.

The read ports are partitioned into fast/slow ports. In 45nm processing technology, the fastest 60% to 80% of ports can be classified as fast ports and correspondingly, the slowest port in the slow ports requires 1.16 to 1.22 cycle time to complete a read access [13]. Since they are assigned two cycles for the read operation, at least 78% of the cycle time can be used to tolerate the extra delay caused by NBTI degradation. Therefore, aggressively using the slow ports will not affect the VL frequency nor, as a consequence, the required guardband.

Note that the access time also varies among fast ports and there is a fraction of fast ports with short access times which allow them to be continuously utilized (their PMOS are under the stress mode) without contributing to the NBTI guardband. We define them as absolute fast ports (AFPs). The remaining fast ports are called possible fast ports (PFPs) because the NBTI degradation on them likely leads to a time violation and contributes to the NBTI guardband. We estimated the read port speed of each RF entry across 400 chips under the impact of PV and observed that on average the fastest 36% read ports in a chip can be classified as AFP since they are at least 15% faster than the VL cycle time. One may notice that even using AFP we may still eventually fail to meet the time specification since NBTI degradation can cause as much as 20% frequency loss during the targeted lifetime period [8]. The PFP still needs to be used in case there is no available AFP. Meanwhile, using PFP lowers the threshold for AFP classification and increases the fraction of ports that can be included in the AFP category. As a result, the overall guardband requirement should consider the wear-out of both PFP and AFP and is determined by the maximum of the two. Migrating RF port utilization from PFP to AFP and slow ports can greatly reduce the guardband requirement. To better understand the proposed technique, we present cycle time variation under the impact of NBTI and PV in Figure 4. Figure 4 (a) shows the baseline case and the optimized scenario is shown in Figure 4(b). In both cases, the read ports are arranged based on their access delay. In the baseline case, the initial cycle time is determined by the longest port delay due to the PV. Generally, NBTI degrades the ports evenly and the final cycle time is an accumulated effect of the worst case in PV and NBTI. On the other hand, with O1, the initial cycle time is greatly improved by VL; the read ports are partitioned into AFP, PFP and slow ports based on their delay and only PFP are vulnerable to NBTI effects. Moreover, NBTI degrades ports unevenly based on their category under the control of O1. Therefore, the cycle time is efficiently reduced compared to the baseline case. The description above mainly focuses on the combinational circuits in RF since it is crucial to the RF frequency. The inversion method proposed in [8] is applied to the SRAM based RF entries for NBTI recovery.

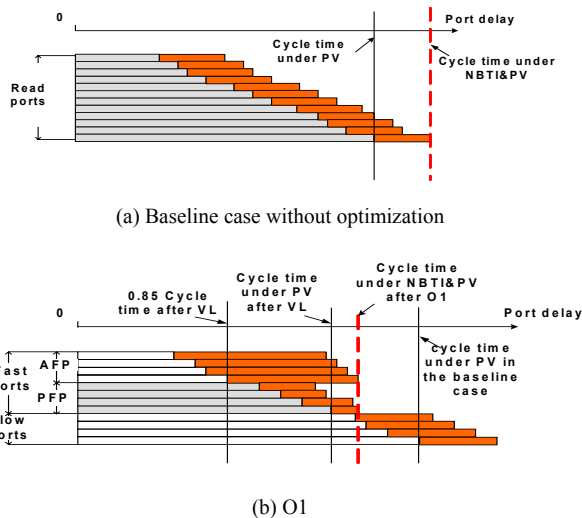


Figure 4. Cycle Time under NBTI and PV Effects

```

1. Every cycle
2. {
3.   IPC update every 100 cycles();
4.   IF (last interval IPC <=1) THEN
5.   {
6.     switch from PFP to slow ports;
7.   }
8.   ELSE
9.   {
10.    IF (AFP is available for switch) THEN
11.    {
12.      switch from PFP or slow ports to AFP;
13.    }
14.    ELSE IF (slow ports is unavoidable) THEN
15.    {
16.      switch from PFP to slow ports;
17.    }
18.    ELSE
19.      no port switching;
20.  }
21. }

```

Figure 5. Pseudo Code for Port Switching in O1

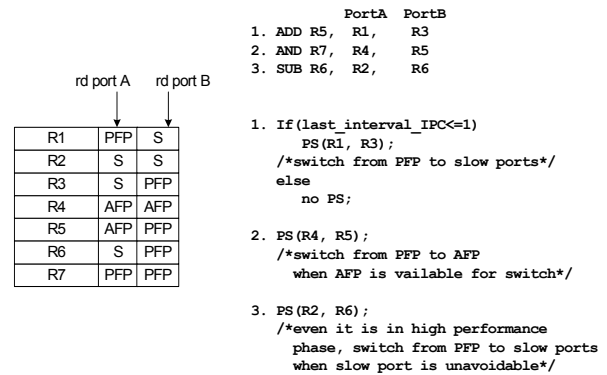
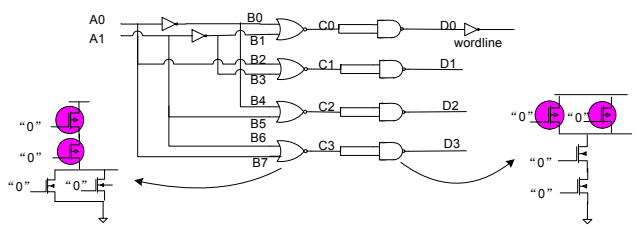


Figure 6. Examples of PS in O1

To implement O1, a key issue is the port utilization assignment. In our proposed scheme, PS is applied to switch from PFP to either AFP or slow ports whenever possible, occurring once the instruction is dispatched into the issue queue (IQ). Since instructions have to stay in the IQ for wakeup and selection, the port information checking and switching can be performed simultaneously without affecting the performance. When the IPC is low, switching from PFP to slow port occurs. The amount of required issue bandwidth is usually low during the low IPC phase and pipe stalls caused by the slow port will cause few issue stalls in the following cycles and hence the impact on performance is small. Intuitively, to avoid the large number of pipe stalls, one needs to limit the number of instructions using slow ports for RF reading. We found that it is unnecessary to do so since there are only about 20% slow ports, the probability that all instructions will be issued on the same cycle, causing pipeline stalls, is low. When the IPC is high, O1 checks the possibility of switching from PFP to AFP. If it cannot be performed and the use of slow port is unavoidable, O1 will try to use a slow port for the other operand read. Because a pipe stall will occur, the performance impact is the same no matter if only one or both of the read ports are slow. However, the NBTI effect is different when one PFP and one slow port are used compared with two slow ports being used. Figure 5 shows the pseudo code of PS in O1. The IPC is updated every 100 cycles and an IPC of 1 is used as a threshold between high and low performance phases. Figure 6 shows an example of port switching in O1. The port information is attached to each register file entry and the operand in each instruction is originally assigned a read port.

The detailed operations are shown when a PS occurs for a given instruction. The implementation of port information profiling and reading, and the hardware support for port switching can be found in [13]. As discussed in [13], VL+PS results in 2% area overhead, O1 introduces extra 1% area overhead to record the port information.

Note that each read port is assigned to a decoder for the port activation. The port is linked to a specific decode line in the decoder. Since the read critical path delay includes the decode delay [13] as well, the NBTI effect caused by port utilization on the decoder cannot be ignored. For illustration, we consider the 2-to-4 decoder in Figure 7. The decode line contains an inverter, a NOR gate, and a NAND gate which also have PMOS transistors. In order to understand the input of each gate for NBTI degradation analysis, a truth table is included in Figure 7. An output of “0” in $D_0 \sim D_3$ causes the port connected to the decode line to be activated for a read operation. In addition, the detailed circuit of NOR and NAND gates are presented to illustrate each PMOS transistor’s stress or recovery mode depending on the two inputs. We show an example where both of the inputs to the gate are “0”. As can be seen, the input “0” stresses the PMOS gate and the input “1” will recover the PMOS. As the truth table shows, when a port is activated, its corresponding decode line will have two “0” inputs in the NOR gate and two “1” inputs in the NAND gate. Correspondingly, the two PMOS transistors in the NOR gate are under stress mode while those in the NAND gate are under recovery mode. When a port is deactivated, there are three input combinations to the NOR gate, which result in either one of the PMOS being under recovery or two of them being under recovery. Additionally, the two PMOS transistors in NAND are under stress mode. Approaches such as resizing transistors [16] can be used to tolerate the NBTI degradation on the inverter, which is not private to a specific decode line. Generally, half of the PMOS transistors in the decode line are under stress mode and the remaining are under recovery mode whenever the port connected to the line is enabled or disabled. In another words, O1 does not affect the amount of NBTI degradation stressed on the decode line. The idea of inserting input vectors [8] when the decoder is idle is used to recover NBTI degradation, solving the uneven degradation problem in the decoder line.



A0	A1	B0	B1	B2	B3	B4	B5	B6	B7	C0	C1	C2	C3	D0	D1	D2	D3
0	0	1	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0
0	1	1	0	0	0	1	1	0	1	0	1	0	0	1	0	1	1
1	0	0	1	1	1	0	0	1	0	0	0	1	0	1	1	0	1
1	1	0	0	1	0	0	1	1	1	1	0	0	0	0	1	1	1

Figure 7. A Example of 2-to-4 Decoder

Another important multi-ported microarchitecture structure in microprocessors is the IQ, which performs out of order

issue of instructions. O1 can be extended to the IQ for PV-aware NBTI mitigation: the CAM read ports (which are used for instruction wake-up and in the critical path) can be partitioned into fast and slow categories. Fast CAM ports are at least 15% faster than the slowest CAM port and they can tolerate NBTI degradation. Techniques similar to O1 can be applied to avoid the use of slow CAM (e.g. attempting to dispatch instructions into the IQ entry with fast CAM, switching the operand from slow CAM to fast CAM when there is only one non-ready operand). We leave a detailed investigation as our future work.

3.3 PV-aware NBTI Mitigation for Combinational Blocks

In this Section, we propose PV-aware NBTI tolerant schemes that target microprocessor combinational blocks. We illustrate our design on the functional units.

As described in Section 2.1, the NBTI recovery in a functional unit can be performed whenever the functional unit is idle. A longer idle time provides more opportunity for NBTI recovery [8], resulting in reduced NBTI guardband. In high performance 64-bit microprocessors, many operand values in the applications do not require the full 64-bit width. These operands are referred to as narrow-width values. When there is an instruction whose operands are narrow-width values, the instruction requires an add operation and the two values only occupy 16 bits. 1/4 of the 64-bit functional unit will be devoted to the instruction’s computation and the remaining 3/4 of the unit can stay in idle mode, providing opportunities for NBTI recovery. As can be seen, narrow-width values can help exploit idle time within a functional unit for NBTI recovery. Previous studies show that there are a large number of narrow-width operations in general purpose applications. For example, in SPEC 2000 INT benchmarks, about 50% of the instructions contain operands no wider than 16 bits. In our study, a 64-bit functional unit is partitioned into four segments with granularities of 16 bits. Each segment can complete 16-bit executions independently. For normal-width values, which are wider than 16 bits, all four segments are involved in computation.

In order to achieve high performance, the combinational blocks in functional units are either pipelined or parallelized. Take the carry look-ahead adder (CLA) as an example. Instead of waiting for the carry to ripple through all the previous stages to find its proper value, as in a ripple carry adder (RCA), the CLA calculates the dependence of each carry-out bit on the first carry-in bit, and parallelizes the carry-out bit computation. Therefore, the add operation in CLA is much faster than in RCA. The frequency of CLA is determined by the longest carry-out bit computation. The disadvantage of CLA is the rapidly increasing complexity as the number of bits increases. A multi-level CLA is proposed to create a larger adder. The frequency of a multi-level CLA is determined by the carry-out computation delay across all the levels. For instance, a 64-bit adder can be built upon 4 parallelized 16-bit CLAs, which match the segment partition introduced above. The 64-bit CLA (partitioned as 4 segments) delay is dominated by the carry-out computation delay in the 16-bit CLAs. The case is similar for other pipelined or parallelized units. As can be seen, the functional units’ frequency is highly related to the critical path delay in each pipelined stage or parallelized block, which is the partitioned segment in our study.

Due to the effect of PV, the critical path delay varies in each segment. The narrow-width operations should not be assigned randomly to the segment without considering the interaction between NBTI and PV. For example, the benefit of narrow-width operations for NBTI guardband reduction will be nullified if the operation is always performed on the segment with the longest delay, which results in more $+\Delta V_{th}$ & $high_ \Delta V_{th}$ cases. Even though other segments achieve high NBTI mitigation, it is equivalent to the case without narrow-width detection since the guardband is determined by the worst-case delay. In this paper, we propose optimization technique 2 (O2) which steers the narrow-width operation to the fastest segment. In general, a functional unit is more resilient to PV than RF because its critical path is longer than that in RF and the delay difference among the segments is usually smaller than 20% [13]. This differs from the AFP in RF since an absolute fast segment is usually nonexistent. The

initial fastest segment will become the bottleneck for the guardband reduction if it keeps being utilized. An online detection of the aggregated effect of NBTI and PV is required to guide migration of the narrow-width operations to the current fastest segment. IDDQ, which describes the standby leakage current in the circuit, can be applied to detect the effect. IDDQ is originally used for testing manufacturing faults [21]. The IDDQ values can demonstrate the underlying parameter variations [22]. Recently, [23] discovered that IDDQ can be applied in NBTI degradation detection as well because the leakage current decreases exponentially as V_{th} increases in transistors. Therefore, IDDQ has the capability to capture both the static and dynamic variations in V_{th} . In our study, the segment with the highest IDDQ is the fastest one and is selected for the narrow-width value operation.

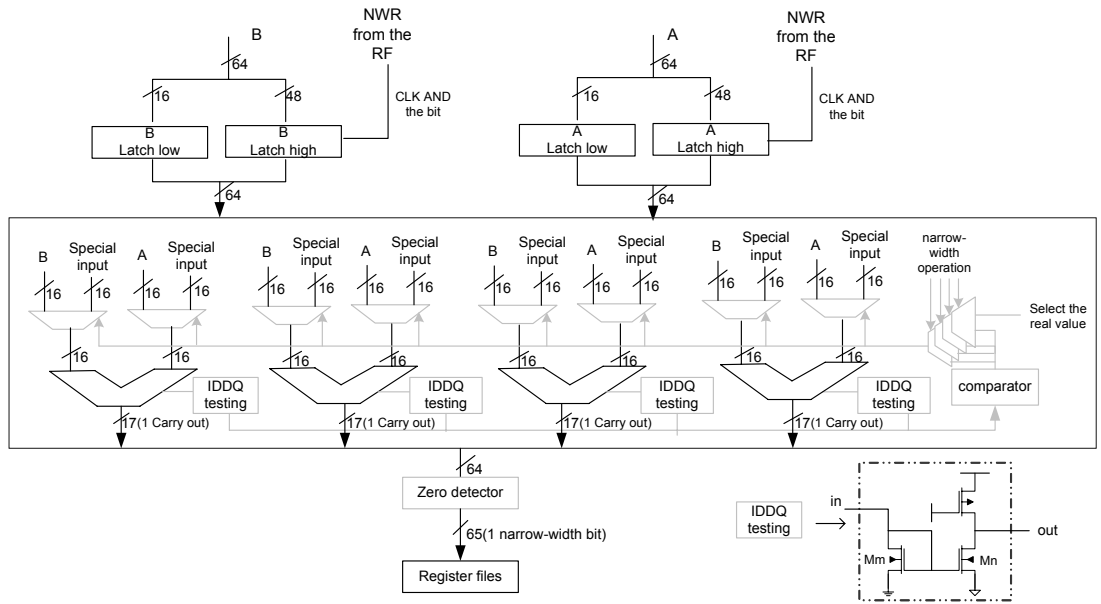


Figure 8. O2 Circuit Design

Figure 8 shows the hardware implementation to support O2. The narrow-width value detection occurs after the result is computed. The 48 most significant bits are checked, in parallel, to determine if they are all 1's (one-detector) or 0's (zero-detector) – indicating that the operand is a narrow-width value. One bit, the narrow width record bit (NWR), is added into the RF entry to record whether the value is narrow width. When the two operands are read-out from the RF and written into the latch, the NWR is checked to determine whether a narrow-width operation can be performed. If it is narrow width, the highest 48 bits will not be latched and will be written directly into the result lines. For each operand, 4 MUXs are added between the latch and the four segments and are used to select an input value to the segment between the NBTI recovery patterns (shown as the special input in Figure 8) and the real value (shown as A or B in Figure 8). Therefore, a total of 8 MUXs are used for the two operands. If this is a narrow-width operation, 4 copies of the 16-bit value (a total of 8 copies for the two operands) will be sent to the MUXs. Otherwise, the

normal value is used as the input. It is possible that the 16-bit operation causes an overflow. In this case, 4 carry-out lines are added in the output. In O2, the IDDQ testing is performed in each segment periodically, the testing current is sent to the 4-input comparator. The comparison output will determine which segment should be selected for the narrow-width operation and its two inputs will be the 16-bit values. Other segments will be inserted with the recovery vector. Another 4 MUXs are added at the output of the comparator before the comparison result is sent to those 8 MUXs for input selection. Because the comparison output should be masked if the current operation is not narrow-width, all of the input should be the real value instead of the NBTI recovery vector. The signal “select the real value” will be multiplexed with the comparison output and the signal “narrow-width operation” determines which signal will be sent out to the 8 MUXs. Similarly, the signal is sent to the output of each segment to decide whose computation result is valid for launching into the result line. The circuit of IDDQ testing, which mirrors the

circuit IDDQ (“in” in Figure 8) to M_n through M_m , is also shown. The analog voltage signal (“out” in Figure 8) reflects the changes in circuit IDDQ. Note that the IDDQ testing and comparator are not in the critical path and they do not introduce any extra delay in the cycle time. As shown in Figure 8, O2 only introduces the MUX and zero detection into the critical path, considering the comparatively long execution path in the functional unit, their effects to the cycle time is negligible. Moreover, their area and power overhead is around 1%.

3.4 PV-aware NBTI Mitigation for Storage Cell Based Structures

In this section, the PV-aware NBTI mitigation technique is proposed for cache, which is the representative storage cell-based structures.

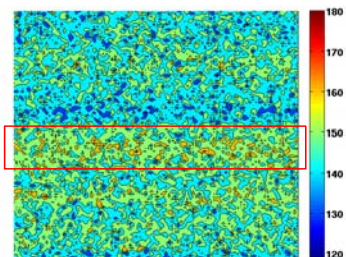


Figure 9. V_{th} (in mV) Variation Map for a Cache

PV exhibits both random and systematic effects. Due to systematic effects, transistors share similar parameters with other (e.g. nearby) transistors. These transistor groups define an area in which the transistors exhibit similar behavior. Since the parameter variation between two transistors is larger as their adjacency distance increases [24], transistors which are far away from this area will exhibit different behavior. If they share another parameter with transistors around them, those transistors can be classified into another area. Figure 9 shows the V_{th} variation map for a cache. As can be seen, the V_{th} variation in cache is not entirely random. Since the cache occupies a large portion of the chip area, transistor V_{th} can be generally high/low in some areas of the cache. Areas with similar V_{th} can be easily found. For other structures, such as RF and functional units, which occupy a small area of the chip, the critical path variation is mainly caused by the random effect since the similar systematic effect performs across the entire structure. Therefore, although the critical paths in the structure are very close, they still vary in the path delay.

It is well known that body biasing (BB) is an efficient method for PV mitigation. However, it must be applied at the structure level and a finer granularity is not achievable with BB technology [14]. Usually, a cache is assigned one BB generator and a uniform voltage biasing is applied in all areas, whether they have high or low V_{th} . The amount of BB applied is determined by the worst case across the entire cache. [8] proposes a NBTI recovery mechanism for cache structures by invalidating 50% of the cache lines and uses them to store the inverted values. However, keeping half of the cache

invalidated increases the cache miss rate and degrades performance, especially on applications that have high cache utilization. When combining the BB technology [14] with the NBTI recovery approach [8], the guardband is reduced significantly. Note that areas with low initial V_{th} can tolerate more NBTI degradation and, as long as the final V_{th} does not exceed that in the areas with high initial V_{th} due to PV, the strict cache line inversion percentage (e.g. 50%) can be appropriately relaxed in those areas. Doing so reduced the number of invalidated cache lines, which decreases the cache miss rate and performance loss, leading to an improvement in the technique efficiency to NBTI and PV mitigation in terms of performance, power, and chip lifetime.

Based on the above observation, we propose O3 to take advantage of the systematic effect of PV in guardband reduction while maintaining performance. We apply adaptive body biasing (ABB) in O3 to mitigate the PV effect. First, O3 partitions the cache into several areas according to the similarity of transistors’ V_{th} . Each area has its individual inversion percentage (areas with lower V_{th} will be assigned a lower inversion percentage, corresponding to a smaller number of invalidated and inverted cache lines). The percentage is estimated based on the difference between the highest V_{th} in the cache and that in the area. Similar to the proposal in [8], the valid/state bits are used to indicate whether the cache line is valid and non-inverted, or invalid and inverted. A counter is used in each area to count the number of inverted cache lines. Once it is below the pre-defined threshold, one LRU cache line is invalidated and written with the inverted value. Since different cache ways are implemented close to one another, the PV exhibits a stronger systematic effect in the horizontal direction than in the vertical direction [25].

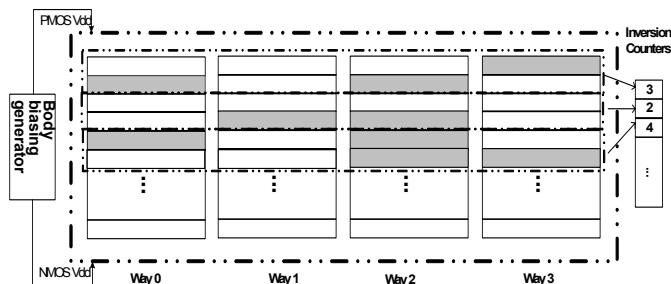


Figure 10. Fundamental Idea of O3 in the 4-way L1Cache

The cache area is partitioned at the set level. However, the partition granularity should be considered. If it is too small, there are fewer cache lines being chosen from the area for inversion and it becomes more difficult to match the required the inversion percentages to a concrete inversion number. In addition, a large number of counters are required for the inversion percentage control, which causes a higher area overhead. On the other hand, if the granularity is too large, the systematic effect cannot be efficiently exploited. For example, when the granularity is set as the entire cache, O3 will be the same as combining BB with the technique proposed in [8]. We perform the sensitivity analysis in Section 5 and choose the

granularity as 8 sets. Figure 10 describes the idea of O3 in the 4-way L1 cache. The cache line with gray color represents the invalidated and inverted lines.

4 EXPERIMENTAL METHODOLOGY

In this Section, we first describe the circuit level experimental methodology, which presents a model of process variation. We then introduce the architecture level evaluation methodology.

4.1 Circuit Level Experimental Methodology: Process Variation and NBTI Modeling

We model variations on L and V_{th} since they are the two major process variation sources [10]. L and V_{th} in each device can be represented as follows:

$$L = L_{nom} + \Delta L_{D2D} + \Delta L_{WID} \quad (\text{Eq. 2})$$

$$V_{th} = V_{th_{nom}} + \Delta V_{th_{D2D}} + \Delta V_{th_{WID}} \quad (\text{Eq. 3})$$

where L_{nom} and $V_{th_{nom}}$ are the nominal value of gate length and threshold voltage respectively. ΔL_{D2D} and $\Delta V_{th_{D2D}}$ represent the D2D variations. Devices in a single die share the same ΔL_{D2D} and $\Delta V_{th_{D2D}}$, which are generally constant offsets. ΔL_{WID} and $\Delta V_{th_{WID}}$ depict the WID variation which can be further expressed as the additive effect of systematic and random variations. We focus our PV modeling on WID variation since the D2D effect can be modeled as an offset value to all the devices in the chip.

To model the random effects of WID variation, we generate random variables that follow a normal distribution. To model systematic variations, we use the multi-level quad-tree partitioning method proposed in [26], which has been widely used in previous PV related work [13, 27]. In this paper, an area of 32 6T SRAM cells is chosen to be the granularity of the smallest quadrant, which is sufficient to describe systematic variation [27]. The WID variation follows a normal distribution (random variables are generated through Monte-Carlo simulation) with standard deviation $\sigma = \sqrt{\sigma_{rand}^2 + \sigma_{sys}^2}$, where σ_{rand} and σ_{sys} depict standard deviations for random and systematic variation respectively. In this study, we simulate processors developed using 45nm process technology and assume $\sigma / \mu = 12\%$ and $\sigma_{rand} = \sigma_{sys} = \sigma / \sqrt{2}$ based on variability projections from [28]. Our baseline machine is an Alpha21264. We scale down the layout from an Alpha21264 chip floor plan to 45nm and generate 400 chips for statistical analysis. Predictive Technology Models [29], the evolution of previous Berkeley Predictive Technology Models (BPTM), are used to provide the basic device parameters for HSPICE simulations. We model the dynamic NBTI degradation in V_{th} by applying the reaction-diffusion (RD) model proposed in [5], the PMOS stress and recovery cycles are obtained via the microarchitectural simulator, and the signal possibility is computed and inserted into the model to determine the shift in V_{th} due to NBTI.

4.2 Architecture Level Evaluation Methodology

We perform detailed architecture simulation using the sim-alpha cycle-level simulator. Additionally, we port Wattch [30] into the simulation framework for dynamic power evaluation, and HotLeakage [31] is used for leakage power estimation. The power results are scaled based on technology projections from ITRS [32]. We use a default Alpha21264 machine configuration with 20-entry INT and 15-entry FP IQs, an 80-entry ROB, an 80-entry INT and 72-entry FP register file with 4-rd/2-wr ports, a 32KB 4-way L1 D-cache, a 32KB L1 I-cache, and a 2MB L2 cache. The processor pipeline bandwidth is set to four. We choose 20 SPEC CPU 2000 integer and floating-point benchmarks. We use the Simpoint tool [33] to identify the most representative simulation interval for each benchmark and each benchmark is fast-forwarded to its representative interval before detailed simulation takes place. We simulate 400 million instructions for each benchmark and present the average result across the 400 chips. In O1, we apply 70% VL technique in RF, which generally obtains a 20% frequency increase compared to the chip without VL-RF. Since both NBTI and PV effects are addressed in our study, we extend the *NBTIefficiency* metric to *NBTI & PV_efficiency* (Eq.4), which quantifies the technique efficiency to both NBTI and PV. Correspondingly, the NBTI+PV guardband is named as *NBTI & PV_guardband*.

$$NBTI \& PV_efficiency = (Delay \cdot (1 + NBTI \& PV_guardband))^3 \cdot TDP \quad (\text{Eq. 4})$$

5 EVALUATION

In this Section, we evaluate the three techniques proposed in Section 3.

5.1 Effectiveness of O1

We compare O1 with the baseline case without any optimization. We also compare the technique combining 70% VL with port switching (PS) and the NBTI mitigation technique, which inserts a special input vector (SIV) in the idle time (we define it as VL+PS+SIV). Figure 11 (a)-(c) presents the CPI, NBTI guardband, and NBTI&PV_efficiency of the three cases in RF. The CPI and NBTI guardband are normalized to the baseline case. The TDP of VL+PS+SIV and O1 is 1.02 and 1.03 respectively due to the area overhead. As shown in Figure 11 (a), CPI increases in both of the NBTI&PV mitigation techniques because the use of slow read ports cannot be eliminated. When they are selected for RF read operation, pipe stalls occur and degrade the performance. However, the performance penalty is negligible in some applications (e.g. *equake, mcf*) because they are running in low IPC phases most of the time and the pipe stalls are tolerated by the low bandwidth requirement. One may notice that O1 increases the CPI by 2% compared to VL+PS+SIV. This happens because slow ports are intentionally chosen for read operations when the IPC is low in order to reduce the PFP utilization. When the IPC information obtained from the last phase generates an incorrect prediction, a slow port is selected by mistake, which causes performance loss. Even though O1 slightly increases the CPI, it gains a significant NBTI guardband reduction. As Figure 11 (b) shows, on average, O1 reduces NBTI guardband by 35% and 36% compared to the baseline case and VL+PS+SIV, respectively. Interestingly, the VL+PS+SIV exacerbates the NBTI degradation compared to the baseline

case because fast ports are used aggressively in VL+PS+SIV and they must accept the utilization migrating from the slow ports. Meanwhile, the SIV does not help reduce the NBTI degradation in read ports since the port switches to the recovery mode automatically when it is free, additionally, the positive effect caused by SIV on the decoder line is not noticeable enough to combat the negative effect. Due to space limitations, we forgo a presentation of NBTI&PV guardband, which is equal to the sum of NBTI and PV guardband. In the baseline case, on average across all the simulated chips, the PV guardband is set to be 0.3, when applying VL technique, improving the frequency by 20% and the PV guardband reduces to 0.1. Figure 11 (c) proves that O1 reduces NBTI&PV_efficiency greatly. It reduces the efficiency as high as 1.00 compared to the baseline case, which implies it improves the efficiency 100% since the best technique has the efficiency of 1 (no PV and NBTI effect). Moreover, it exhibits much stronger ability than VL+PS+SIV in solving NBTI and PV because it achieves 30% improvement in NBTI&PV_efficiency.

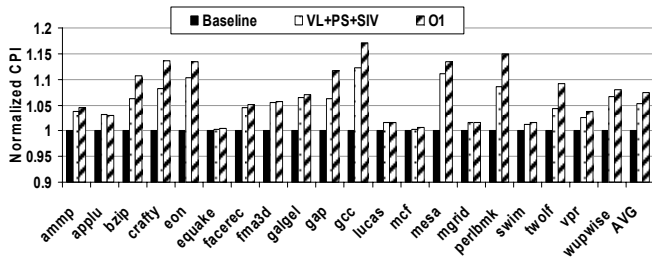


Figure 11. (a) Normalized CPI in RF

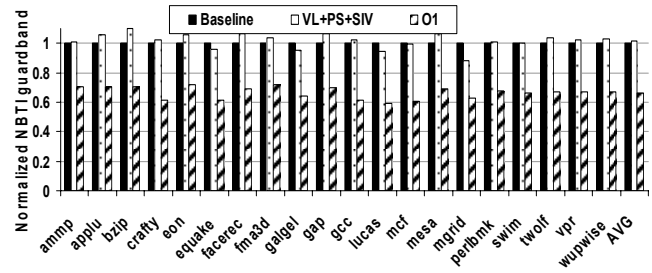


Figure 11. (b) Normalized NBTI Guardband in RF

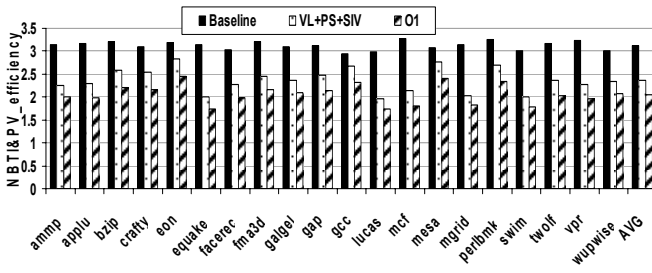


Figure 11. (c) Normalized NBTI&PV_efficiency in RF

5.2 Effectiveness of O2

We compare O2 with the baseline case, the NBTI mitigation technique SIV, the technique which applies SIV and takes narrow-width operation into consideration (define as

SIV+NW). Since the VL technique [13] is orthogonal to the above methodologies, we skip the discussion on their combination to VL due to space limitations. Figure 12 a-b presents the NBTI&PV_guardband, which is normalized to the baseline case, and the efficiency of the four cases in Integer ALU. CPI is not shown in the figure since it has a negligible effect on performance. The TDP in SIV+NW and O2 is 1.01 and 1 in SIV and the baseline case. We show the results of IntALU because most of the narrow-width operations are integer arithmetic and logic operations. It is not fair to judge the efficiency of the techniques in functional units (e.g. FPU) with few narrow-width operations. As Figure 12 shows, compared to the baseline case, on average across all the benchmarks, SIV reduces the guardband by 28%. It gains less reduction than that reported in [8] (63%) because we study the IntALU, which performs both arithmetic and logic operations and has less idle time than the adder studied in [8]. O2 exhibits much stronger capability in guardband reduction, which are 55% and 59% in INT and FP benchmarks respectively and, as a result, improves the efficiency by 73% and 76% in the two benchmark categories. Compared to SIV+NW, which blindly assigns the narrow-width operations inside the unit, O2 decreases the guardband 15% and 12% in INT and FP benchmarks. This contributes to 18% and 13% efficiency improvement compared with SIV+NW.

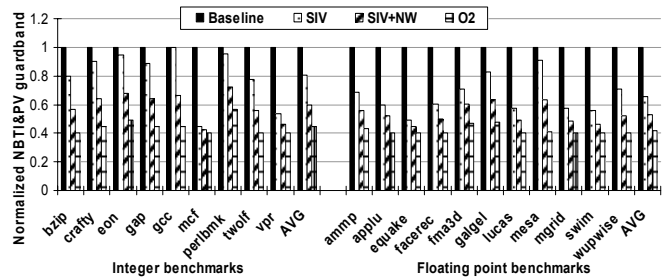


Figure 12. (a) Normalized NBTI&PV Guardband in IntALU

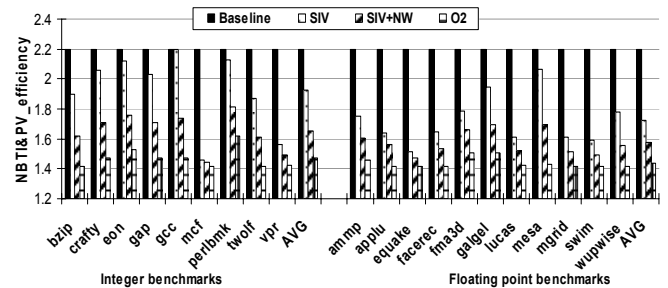


Figure 12. (b) Normalized NBTI&PV_efficiency in IntALU

5.3 Effectiveness of O3

Figure 13 (a)-(b) shows the normalized CPI and NBTI&PV_efficiency generated by the baseline case, the technique applying ABB with cache line inversion (CLI) (define as ABB+CLI), and O3. Since the NBTI and PV problem can be easily solved in the L2 cache by implementing periodical inversion [34], we focus our study on L1 data cache. Note that HotLeakage is used to evaluate the power overhead caused by ABB. As can be seen, ABB+CLI has negligible CPI impact on some benchmarks (e.g. lucas, mcf) because of frequent L2 cache misses: a L1 miss latency caused by the

cache line inversion will be covered by the L2 miss which occurs simultaneously. However, it degrades the performance significantly on benchmarks with low L2 cache miss rates. O3 solves this problem since it efficiently utilizes the L1 resources. For example, O3 improves the performance by 19% in *eon* and 8% in *mesa*. As shown in Figure 13 (a), O3 obtains similar CPI results as the baseline case. It improves the NBTI&PV efficiency 13% compared to ABB+CLI. Figure 14 describes the NBTI&PV_efficiency obtained by O3 as the granularity varies from a single set to the entire cache. We perform the analysis on benchmarks (e.g. *eon*, *vpr*) which are sensitive to ABB+CLI technique. As expected, the performance loss is high when the granularity is extremely small or large. An 8-set granularity achieves the best efficiency, it is chosen in the O3 implementation but requires an extra cache line and 16 counters, which results in 1% additional area overhead.

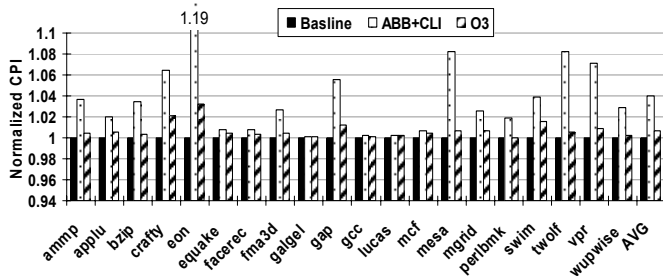


Figure 13. (a) Normalized CPI in L1 Cache

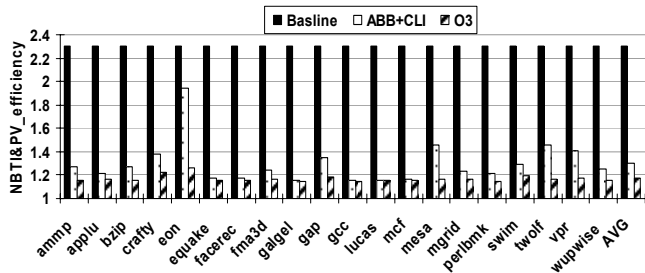


Figure 13. (b) Normalized NBTI&PV_efficiency in L1 Cache

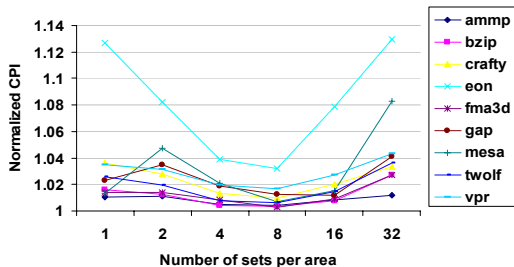


Figure 14. NBTI&PV_efficiency with Various Granularity

5.4 NBTI&PV Efficiency Regarding to the Entire Chip

In order to evaluate the effectiveness of the three proposed techniques on the entire chip, we compute the NBTI&PV_efficiency of the processor following the equations proposed in [8], based on each structure's Delay, NBTI&PV_guardband, and TDP generated by our techniques. On average, we obtain an efficiency of 2.20 for the entire chip.

In the baseline case without any optimization, the chip NBTI&PV_efficiency goes up to 3.375. As can be seen, our techniques improve the efficiency by 117%. The effectiveness of simply combining PV and NBTI mitigation techniques is evaluated for the comparison, its NBTI&PV_efficiency is 2.41, and our technique outperforms this technique by 21%.

6 RELATED WORK

There have been several studies on NBTI modeling and mitigation at both the circuit and microarchitectural levels. The Reaction-diffusion (R-D) model has been widely used to model the NBTI degradation and recovery effect [4, 6]. [5] recently considered temperature variation in the NBTI model. The impact of NBTI on the performance of combinational circuits is investigated in [35], which shows that NBTI degradation is sensitive to the input patterns and the stress time. In addition, the NBTI effect on SRAM array is modeled and studied in [36], where it is shown that the read stability degrades due to NBTI and that the degradation is exacerbated in the presence of PV. To mitigate combinational circuit aging under NBTI, adaptive body biasing (ABB) is applied in NBTI resilient circuits [37]. [7] proposes to identify the critical gates that are most important for timing degradation and protects them from NBTI. To improve the storage cell reliability under NBTI, [38] proposes a new memory cell design consisting of a number of NAND gates instead of inverters to reduce the average degradation on each PMOS. Periodic inversion [34] is proposed to flip the contents of all cells periodically, keeping the balance between “0” and “1” in the cell and is an efficient way to mitigate NBTI in storage cells, but the extra flipping delay in the critical path causes 10% frequency loss. [39] improves the cache reliability under NBTI. It proposes proactive use of microarchitectural redundancy, in which the two components operate either in active mode or in recovery mode, periodically transitioning between the two modes according to a recovery schedule. The combined effect of PV and NBTI has been modeled and analyzed in [40, 41]. Moreover, [20] proposes online PV and NBTI detection in logic circuits and applies ABB to tolerate the V_{th} variations. [42] proposed a technique called “Razor” to tune the supply voltage by monitoring the error rate caused by PV and NBTI during circuit operation, thereby eliminating the need for voltage margins. “Razor” mainly targets combinational logics. In our study, we target the mitigation of NBTI and PV effect in both combinational circuits and storage cell based structures with desirable trade-offs among performance, reliability, and power. To our knowledge, this is the first work taking advantage of the interplay between PV and NBTI to efficiently address the variation problem caused by NBTI and PV.

7 CONCLUSIONS

NBTI is a growing concern in nanometer technology. It degrades PMOS transistors by increasing their V_{th} , which leads to failures in both logic circuits and storage cells. Meanwhile, process variations (PV), which result in a static parameter variation (e.g L and V_{th}) in transistors, exacerbate the reliability problem in current high performance processors. Methodologies to mitigate both PV and NBTI effects are highly desired. In this study, we observe that techniques

leveraging the positive interaction between PV and NBTI can obtain attractive trade-offs among performance, reliability, and power. We propose three microarchitecture optimizations to efficiently take advantage of the positive interplay between NBTI and PV to mitigate NBTI effect in the presence of PV. Our techniques are flexible and can be applied to most of the microarchitecture structures. Our experimental results show that the aggregated effect of the proposed methods has the ability to improve the chip NBTI&PV_efficiency by 117% compared to the baseline case without any optimization, and by 21% compared to the technique which simply combines NBTI and PV mitigation methods.

ACKNOWLEDGMENT

This work is supported in part by NSF grants CNS-0834288, CCF-0811611, CNS-0720476, by SRC grants 2008-HJ-1798, 2007-RJ-1651G, by Microsoft Research Trustworthy Computing, Safe and Scalable Multi-core Computing Awards and by two IBM Faculty Awards. José Fortes is also funded by the BellSouth Foundation.

REFERENCES

- [1] D. K. Schroder, and J.A. Babcock, Negative Bias Temperature Instability: Road to Cross in Deep Submicron Silicon Semiconductor Manufacturing. In the Journal of Applied Physics, 2003.
- [2] L. Peters, NBTI: A Growing Threat to Device Reliability, Semiconductor International, 2004.
- [3] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer, High-performance CMOS Variability in the 65-nm Regime and Beyond, IBM J. Res. & Dev., 2006.
- [4] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, An Analytical Model for Negative Bias Temperature Instability, In Proceedings of ICCAD, 2006.
- [5] H. Luo, Y. Wang, K. He, R. Luo, H. Yang and Y. Xie, Modeling of PMOS NBTI Effect of Considering Temperature Variation, In Proceedings of ISQED, 2007.
- [6] R. Vattikonda, Y. Luo, A. Gyure, X. Qi, S. Lo, M. Shahram, Y. Cao, K. Singhal, and D. Toffolon, A New Simulation Method for NBTI Analysis in SPICE Environment, In Proceedings of ISQED, 2007.
- [7] W. Wang, Z. Wei, S. Yang, and Y. Cao, An Efficient Method to Identify Critical Gates under Circuit Aging, In Proceedings of ICCAD, 2007.
- [8] J. Abella, X. Vera, A. Gonzalez, Penelope: The NBTI-Aware Processor, In Proceedings of MICRO, 2007.
- [9] S. Borkar, T. Kamik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De., Parameter Variations and Impact on Circuits and Microarchitecture, In Proceedings of DAC, 2003.
- [10] 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, 2002.
- [11] M. Orshansky, L. Milor, P. Chen, K. Keutzer, and C. Hu, Impact of spatial intrachip gate length variability on the performance of high-speed digital circuits, In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, May 2002.
- [12] H. Chang and S. S. Sapatnekar, Full-chip Analysis of Leakage Power under Process Variations, including Spatial Correlations, In Proceedings of DAC, 2005.
- [13] X. Liang and D. Brooks, Mitigating the Impact of Process Variations on Processor Register Files and Execution Units, In Proceedings of MICRO, 2006.
- [14] R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, Mitigating Parameter Variation with Dynamic Fine-grain Body Biasing, In Proceedings of MICRO, 2007.
- [15] V. Reddy, J. Carulli, A. Krishnan, W. Bosch and B. Burgess, Impact of Negative Bias Temperature Instability on Digital Circuit Reliability, In Proceedings of IRPS, 2002.
- [16] R. Vattikonda, W. Wang, and Y. Cao, Modeling and Minimization of PMOS NBTI Effect for Robust Nanometer Design, In Proceedings of DAC, 2006.
- [17] C. Schlunder, R. Brederlow, B. Ankele, A. Lill, K. Goser and R. Thewes, On the Degradation of P-MOSFETs in Analog and RF Circuits under Inhomogeneous Negative Bias Temperature Stress, In Proceedings of IRPS, 2003.
- [18] W. Abadeer and W. Ellis, Behavior of NBTI under AC Dynamic Circuit Conditions, In Proceedings of IRPS, 2003.
- [19] X. Liang and D. Brooks, Latency Adaptation for Multiported Register Files to Mitigate the Impact of Process Variations, In Workshop on ASGI, 2006.
- [20] K. Kang, K. Kim, and K. Roy, Variation Resilient Low-Power Circuit Design Methodology Using On-Chip Phase Locked Loop, In Proceedings of DAC, 2007.
- [21] R. Rajsuman, IDDQ Testing for CMOS VLSI, Proceedings of the IEEE, 2000.
- [22] A. Agarwal, K. Kang, and K. Roy, Accurate Estimation and Modeling of Total Chip Leakage Considering Inter-&Intra- Die Process Variations, In Proceedings of ISLPED, 2005.
- [23] K. Kang, M. A. Alam, and K. Roy, Characterization of NBTI induced Temporal Performance Degradation in Nano-Scale SRAM array using I_{DDQ} , IEEE International Test Conference, 2007.
- [24] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey and C. Spanos, Modeling Within-die Spatial Correlation Effects for Process-design Co-optimization, In Proceedings of ISQED, 2005.
- [25] S. Ozdemir, D. Sinha, G. Memik, J. Adams, and H. Zhou, Yield-aware Cache Architectures, In Proceedings of MICRO 2006.
- [26] A. Agarwal, D. Blaauw, S. Sundareswaran, V. Zolotov, M. Zhou, K. Gala, and R. Panda, Path-based Statistical Timing Analysis considering Inter and Intra-die Correlations, In Proceedings of TAU, 2002.
- [27] K. Meng, and R. Joseph, Process Variation Aware Cache Leakage Management, In proceedings of ISLPED, 2006.
- [28] A. Kahng, The Road Ahead: Variability. Design & Test of Computers, 2002.
- [29] NIMO Group, Arizona State Univeristy. PTM homepage. <http://www.eas.asu.edu/~ptm/>.
- [30] D. Brooks, V. Tiwari and M. Martonosi, Watch: A Framework for Architectural-Level Power Analysis and Optimizations, In Proceedings of ISCA, 2000.
- [31] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects, Technical Report CS-2003-05, University of Virginia, 2003.
- [32] International Technology Roadmap for Semiconductors (2006 Update).
- [33] T. Sherwood, E. Perelman, G. Hamerly and B. Calder, Automatically Characterizing Large Scale Program Behavior, In Proceedings of ASPLOS, 2002.
- [34] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, Impact of NBTI on SRAM Read Stability and Design for Reliability, ISQED, 2006.
- [35] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, Frank Liu and Y. Cao, The Impact of NBTI on the Performance of Combinational and Sequential Circuits, In Proceedings of DAC, 2007.
- [36] K. Kang, H. Kufluoglu, K. Roy, and M. A. Alam, Impact of Negative-Bias Temperature Instability in Nanoscale SRAM Array: Modeling and Analysis, IEEE Trans. on CAD, 2007.
- [37] Z. Qi and M. Stan, NBTI Resilient Circuits Using Adaptive Body Biasing, In Proceedings of GLSVLSI, 2008.
- [38] J. Abella, X. Vera, O. Unsal and A. González, NBTI-Resilient Memory Cells with NAND Gates for Highly-Ported Structures, In Workshop on DSN, 2007.
- [39] J. Shin, V. Zyuban, P. Bose, and T. Pinkston, A Proactive Wear-out Recovery Approach of Exploiting Microarchitectural Redundancy to Extend Cache SRAM Lifetime, In Proceedings of ISCA, 2008.
- [40] S. Basu and R. Vemuri, Process Variation and NBTI Tolerant Standard Cells to Improve Parametric Yield and Lifetime of ICs, In Proceedings of ISVLSI, 2007.
- [41] Th. Fischer, A. Olbrich, G. Georgakos, B. Lemaitre, and D. Schmitt-Landsiedel, Impact of Process Variations and Long Term Degradation on 6T-SRAM Cells, Advances in Radio Science, 2007.
- [42] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner and T. Mudge, Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation, In Proceedings of MICRO, 2003.