

NBTI Mitigation in Microprocessor Designs

Simone Corbetta, William Fornaciari
Politecnico di Milano – Dipartimento di Elettronica e Informazione
Via Ponzio 34/5, 20133 Milano, Italy
{scorbetta, fornacia}@elet.polimi.it

ABSTRACT

Negative-Bias Temperature Instability seriously affects nanoscale circuits reliability and performance. Continuous stress and increasing operating temperatures lead to device degradation and long-term system unavailability. The opportunity to optimize the duty-cycle of the stress/recovery phases to reduce V_{th} degradation leads to innovative research of reliability-oriented resources allocation at architectural level. This work explores the impact of different allocation strategies on the processor degradation, through a novel estimation methodology. Experimental results show that the proposed NBTI-aware allocation strategy can guarantee from 10% and up to 30% lower degradation compared to classical strategies, under different operating scenarios and under process variability.

Categories and Subject Descriptors

B.8 [Performance and Reliability]: Reliability, Testing, and Fault-Tolerance

General Terms

Design, Reliability

Keywords

Dynamic instruction scheduling, NBTI degradation

1. INTRODUCTION

As microelectronics scales down to 45nm and beyond, the increase in reliability concerns is becoming of paramount importance. Several physical mechanisms that are mining modern electronic devices have been spotted and studied, and are still under investigation [11]. Reliability mechanisms can be classified into two broad categories: interconnect wear-out and device wear-out. Interconnection degradation is mainly due to electromigration, caused by high density in the current flowing through resistive lines [2]; device degradation, on the other hand, are those related to transistor-level failures, such as oxide breakdown and Negative-Bias Temperature Instability (NBTI) [3]. In addition, thermal cycling and

hot-spots represent a true challenge for system reliability. In this perspective the need to optimize for reliability is turning into a hard design constraint. The birth and growth of multi-core architectures provides room to optimize system-level reliability through appropriate design techniques at the micro-architectural, architectural and Operating System levels. As an example, the presence of multiple (redundant) cores benefits from smart scheduling policies, such that thermal and reliability concerns are under direct control. In general, the presence of multiple functional units, as in superscalar processors or in MPSoC devices, allows the designers to optimize the resource allocation with emphasis on the reliability properties of the system. Run-time optimization approaches at the hardware and software level are gaining increasing attention from the scientific and engineering community. Classical power management and thermal management techniques, either DVFS-like or at micro-architecture level (e.g., clock gating), and OS-level techniques as task migration should then consider reliability as a design constraint of paramount importance. Meanwhile, the growth of density in ULSI devices makes it costly to employ early-stage evaluation of the benefits of power/thermal and reliability policies.

Purpose of the work presented in this paper is to address the problem of early-stage evaluation of NBTI mitigation techniques at architectural level, and the mitigation of NBTI-induced degradation through appropriate instruction scheduling. In this perspective, a novel methodology is proposed, and a novel policy for NBTI directed allocation strategy is compared against classical Round-Robin or Random-Scheduling strategies. The proposed strategy is meant to be used in single-core superscalar/VLIW processors or multi-core architectures, in which there generally exist different functional units accommodating the instructions execution.

1.1 Related works

Reliability-oriented management of complex systems is recently gaining interest due to the inherent unreliability of nanoscale devices and VLSI systems [5]; NBTI-aware resource usage is a relatively new concept attaining interest from the researchers, thanks to the particular feature of stress/recovery phases in the NBTI degradation process. Few works have been proposed in this context in the past. The benefits of appropriate Input Vector Control (IVC) mechanisms for NBTI mitigation have been demonstrated recently in [23]. The authors show that IVC techniques provide both mitigation of long-term degradation and leakage power control. Experimental results show that the worst-case NBTI-induced degradation is within 30% of the initial circuit delay, thus reducing the performance degradation along the critical path. For leakage power, this is in the order of 10% [23]. The authors in [18, 17] on the other hand, exploit the recovery phase typical of the NBTI process in order to mitigate degradation. The authors address emerging multi-core architectures, proposing a workload balancing scheme [17]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GLSVLSI'12, May 3–4, 2012, Salt Lake City, Utah, USA.
Copyright 2012 ACM 978-1-4503-1244-8/12/05 ...\$10.00.

that aims at alternating cores between full workload phases and recovery phases. The scheduling of stress and recovery phases is exploited through the concept of “workload capacity” [18], in which the impact of the degradation on the delay is computed at run-time; cores are associated to delay-driven performance metrics, used to rank the cores from their degradation status stand-point.

1.2 Novel contributions

In this work we present a novel methodology to estimate the NBTI-induced degradation at a system-level, taking into account low-level (i.e., transistor- and gate-level) analytical models to compute the NBTI degradation. We apply the proposed methodology to estimate the impact of appropriate allocation strategies at architectural level. Our work differs from the previous ones in several ways. At first we export to upper levels of abstraction the concept of IVC technique, focusing on the impact that instructions execution has on the available functional units of the processor; this is done through the definition of a novel allocation strategy for instructions selection. We then directly address the problem of workload balancing considering the cumulative effects (integrated over time) of the usage of functional units of interest. The rationale stands in the possibility to use recovery phases in a single-core processor without stopping the entire processor, but appropriately alternating the execution of instructions according to the functional units they stress. The proposed methodology is also a general approach that can be used to estimate the worst-case degradation of any hardware design, as a function of the stress duty-cycle in each RTL block of interest.

1.3 Paper structure

This paper has been organized as follows. Section 2 will introduce the reader with basic definitions of the NBTI process, and the degradation induced by threshold voltage shift, along with a brief explanation of the employed analytical models. Section 3 will discuss in details the proposed estimation methodology. Experimental results are then presented and discussed in detail in Section 4. Conclusions are drawn in Section 5.

2. NBTI OVERVIEW

Negative-Bias Temperature Instability (NBTI) is one of the most serious reliability concerns in nanoscale digital devices, gaining increasing attention beyond 45nm [11]. The degradation induced by NBTI is a complex chemical-physical process affecting both performance and availability of PMOS transistors; in these devices, NBTI degradation occurs when the gate-to-source V_{gs} control voltage is negative. This process is mainly due to bond dissociation at the silicon-oxide interface [3]: due to the high electric field, interface traps are created through a complex chemical-physical reaction, moving ions far from the interface. Traps generate charges at the interface, increasing the threshold voltage at long-term stress times. The time-varying change in V_{th} has direct impact on the performance of the transistor and on its availability; indeed, if the degraded V_{th} reaches sufficient magnitude, the PMOS device becomes unable to drive current to the load gates. However, one very important aspect of this degradation process stands in the presence of a recovery phase. Once the stress is removed, i.e. $V_{gs} > 0$, the threshold voltage degradation tends to decrease to a fraction of its original value, thus restoring part of the initial V_{th} value. Unfortunately, the cumulative stress cannot be entirely removed from the device during the recovery phase, but a fraction [3], leading to long-term permanent faults [22]. This aspect is of paramount importance in the mitigation of NBTI degradation, and it effectively acts as the driving force of our research work. As a matter of fact, we can

Table 1: Technology parameters and constant values of the analytical model in Equation 1. Values are chosen according to suggestions and technology models from ITRS [10].

Parameter	Role	Value
q	electric charge	1.602×10^{-19} C
t_{ox}	oxide thickness	1.2 nm at 90 nm
ϵ_{ox}	relative permittivity	3.9 for SiO ₂
K	constant	$8 \times 10^4 \text{ s}^{-0.25} \text{ C}^{-0.5} \text{ nm}^{-2}$
C'_{ox}	oxide capacitance per unit area	$3.136 \times 10^3 \text{ nFcm}^{-2}$
W_{eff}	PMOS geometry W	1×10^4 nm
L_{eff}	PMOS geometry L	45 nm
E_0	reference electric field	2 MVcm ⁻¹
E_a	activation energy	0.49 eV
k	Boltzmann's constant	$8.617 \times 10^{-5} \text{ eVK}^{-1}$
T_0	constant	1×10^{-8}
n	constant for H_2 diffusion	1/6

mitigate at some extent the degradation due to NBTI by controlling the stress and recovery phases duty cycle on the devices: lower duty cycles will result in lower stress for a PMOS transistor, and analytical models demonstrate that lower duty-cycles have a positive impact on the device lifetime [22]. At architectural level, this means that we can mitigate the degradation of a microprocessor by selectively controlling the duty-cycle usage of the functional units, preferring those that are under lower stress; in this way the device lifetime is prolonged. In more details, our work drives the scheduling of instructions in the target processor architecture according to the status of microprocessor functional units usage. The purpose of this work is to study the impact that different instructions allocation policies have on the device lifetime, as opposed to classical policies, e.g. Round-Robin, through the proposed methodology.

2.1 Analytical models

Mathematical models that can be found in literature [2, 3] explain in details the chemical-physical reaction on the basis of the NBTI degradation; however, they apply to transistor-level degradation estimation, and the algorithms for their estimation are very time consuming. Predictive analytical models have been derived for instance in [17, 4, 13] and they address gate-level computation. The degradation estimated in this work is based on analytical models shown in [22] for the long-term degradation computation as a function of the gate-level duty-cycle, and reported in Equation 1.

$$\Delta V_{th}^{t_{ref}} \approx \left(\frac{n^2 K_v^2 \alpha C t_0 t_{ref}}{\xi_1^2 t_{ox}^2 (1 - \alpha)} \right)^n \quad (1)$$

where the parameters K_v and C are defined as in Equation 2a and Equation 2b, respectively.

$$K_v = \left(\frac{qt_{ox}}{\epsilon_{ox}} \right)^3 K^2 C_{ox} (V_{gs} - V_{th}) \sqrt{C} \exp \left\{ \frac{2E_{ox}}{E_0} \right\} \quad (2a)$$

$$C = T_0^{-1} \exp \left\{ -\frac{E_a}{kT} \right\} \quad (2b)$$

The definition of $\Delta V_{th}^{t_{ref}}$ approximates the degradation as a function of the reference wall-time t_{ref} , when the duty-cycle is α . The parameters in Equation 1 are explained in Table 1, whose values are derived both from ITRS technology projections and [10]. The single duty-cycle parameter α captures the effects of stress and recovery phases, to appropriately estimate the long-term degradation.

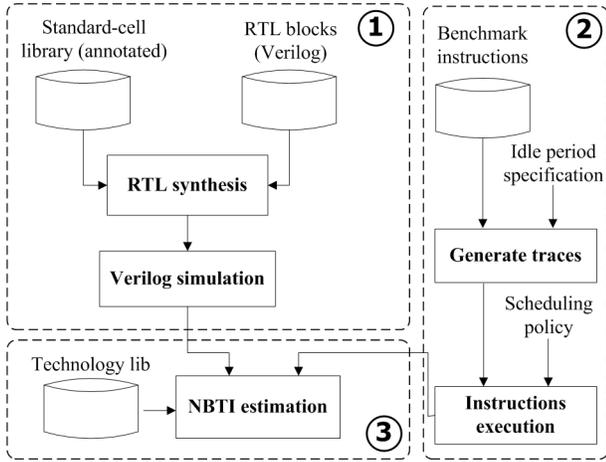


Figure 1: NBTI degradation estimation methodology overview.

3. ESTIMATION METHODOLOGY

The methodology flow is depicted in Figure 1. The proposed methodology develops a worst-case estimate of the degradation induced by the NBTI process at an architectural level, as a function of the duty-cycle usage of the different functional units of the target microprocessor. Starting from a low-level characterization of the functional units in the processor (Stage#1), our methodology emulates the execution of instructions and their impact on the duty-cycle usage of such units (Stage#2); then, the NBTI degradation is estimated in terms of threshold voltage increase ΔV_{th} (Stage#3). To this extent the proposed methodology is actually general enough to be applied to any RTL design of interest.

3.1 Stage#1: RTL blocks characterization

The purpose of this stage is to analyze the behavior of RTL blocks in the reference design (e.g., a processor), as a function of the input vectors. Starting from synthesizable RTL description (either Verilog or VHDL), the hardware design is firstly mapped to a standard-cell library. The *RTL Synthesis* block in Figure 1 computes the gate-level netlist of the design employing an annotated version of the technology library. This augmented version contains detailed information on the number of PMOS transistors that are stressed: for each cell in the standard-cell library, the number of PMOS transistors for which the V_{gs} control voltage is negative (i.e., logical 0) is statically computed. This is done by counting, for each gate in the standard-cell library, the number of PMOS transistors for which the input control voltage is a logical 0, for each input vector; this can be easily done by direct inspection of the gate circuits or through circuit-level simulation. Notice that we are not considering “weak” effects of degradation for stacked PMOS circuits as authors do in [15], since we are interested in worst-case estimation. However, as opposed to the work done in [15], we consider 3-inputs and 4-inputs gates, and not only 2-inputs gates. The mapped design collects information about the stress profile on a per-gate basis; this static information will be used later on in Stage#3 while estimating the NBTI-induced degradation in the reference design. The mapped design is simulated across several iterations, where input vectors are chosen uniformly distributed; notice that the Uniform distribution assumption of input vectors leads to a worst-case estimate, since generally input vectors follow a different pattern (e.g., Gaussian distribution). The output of Stage#1 is detailed information on the (average) number of PMOS transistors that are stressed

in *each* RTL block, such that a static characterization of the functional units has been achieved.

3.2 Stage#2: Instructions emulation

The purpose of this stage is to provide a quantitative statistical characterization of the duty-cycle of each functional unit composing the reference processor, as a function of the instruction execution trace. We select an appropriate suite of diverse benchmarks (refer to Section 4 for detailed information on the selected benchmarks), such that to cover the widest applications scenario with each benchmark bringing information regarding the statistical instructions breakdown. Using an ISS (Instruction-Set Simulator), we compute the percentage of execution of each instruction class and use it as an input to the proposed estimation flow, in the form of *Benchmark instructions* reference data. To generate traces, resembling the real execution of multiple instructions, several instructions are sampled iteratively. To be able to capture the non-continuous workload assignment of real processors (e.g., periodic as well as non periodic tasks), an idle period specification parameter is used to model the idle/active time. This is done in *Generate traces* block, that generates the emulated instructions flow. The *Instructions execution* step chooses instructions from the emulated traces, according to the desired scheduling policy: it is in this stage that the effective duty-cycles are affected by the dynamic instruction scheduling strategy. This last information will be used in Stage#3 to estimate the NBTI-induced degradation.

3.3 Stage#3: NBTI estimation

Stage#3 takes as input the static characterization of PMOS behavior from Stage#1 and the duty-cycle usage on RTL blocks from Stage#2 in order to provide an estimation of the NBTI-induced V_{th} degradation. Analytical models from Section 2 are employed in this part. In particular, the number of PMOS transistors computed from the synthesis in Stage#1 is used to represent the circuit as a bare collection of (PMOS) transistors; this matrix representation is initialized with default threshold voltage values according to the reference technology node and according to process variation parameters (see Section 5 for more details on this). The execution traces and duty-cycle usage from Stage#2 are used to compute the NBTI stress on a subset of the available PMOS transistors: at each iteration (defined by the instructions epoch) a subset of N PMOS transistors is randomly chosen and the degradation models from Section 2 are applied to that PMOS transistors. N is the average number of PMOS transistors for which the input is a logical 0, as computed by Stage#1. At each iteration, the ΔV_{th} value is then updated.

3.4 Selection policies

To compare the impact of the allocation policy on the reliability of the processor, we applied the methodology to different scheduling strategies, and compared the results (refer to Section 4 for quantitative analysis on this). We compared three different scenarios: Round-Robin (RR), Random Selection (RS) and Custom Selection (CS). In the RR policy, scheduling makes use of a circular buffer whence a new set of instructions is selected. Once the associated time quantum expires, a new set is sampled from the pool of available ones. A RR scheduling aims at fairness in the use of the computational resources, but it does not take into consideration the type of instructions and it is not suited for direct duty-cycle control on processors functional units. RS strategy, on the other hand, randomly chooses an application to be executed from the set of available ones, without any auxiliary information but the number of running applications. The proposed CS approach, last, directly

takes into account the type of each instruction, according to the functional unit it requires to be executed, and tries to employ such information to drive the scheduling for reliability optimization.

More formally, we are given a set $P = \{p_1, p_2, \dots, p_{NP}\}$ of NP tasks, each task p_j having M_j instructions to be executed. For simplicity, assume that M_j is known. The reference processor is composed of a set $F = \{f_1, f_2, \dots, f_{NF}\}$ of NF functional units; for instance we can have integer ALUs, floating-point comparators or multipliers. In this work we assume that instructions are ready to execute as they are read from the instruction cache memory. In this way, we will focus only on balancing the use of the resources during the execution stage of the processor pipeline. Each instruction i_j generally requires a single functional unit to be executed, and this can be conveniently specified in a mapping function $E \in P \times F$. Entry e_{ij} is set to 1 if and only if functional unit $j \in F$ accommodates the execution of instruction $i \in P$. During the execution, each task is simply assumed to be a sequence of set of W instructions, representing the number of instructions that are executed (without preemption) in each available time quantum (called the “epoch”). For simplicity, we assume there exists a time quantum of exactly W instructions independently on the scheduling policy; the aim of scheduling is to decide which set should be instructions taken from. At the end of each time slot, the scheduler is invoked once again, and the next benchmark to execute is selected. Since the purpose of our policy is to balance the usage of the functional units, we continuously keep track of the resources required by each instructions window for each available tasks, and use this information to guide the scheduling decision.

The proposed policy aims at load-balancing the stress of each micro-architectural block of interest, in order to span the stress over the entire chip area, and to mitigate the appearing of hot-spots (from a reliability view-point) in a particular chip region. To this extent a suitable metric has been proposed and analyzed (refer to Section 4 for quantitative details on this). Denoting with d_j the duty-cycle usage statistics for the generic functional unit $j \in F$, and with w_i a weight associated with the generic instruction i , starting from the current Program Counter PC , the weighted value for the current epoch is computed, for each available task, as $\xi_j = \sum_{i=PC}^{PC+W} (d_{c_j} \cdot w_i)$. The weight w_i is chosen according to the latency of the functional unit that ensures execution of the generic instruction; generally, instructions can be classified according to either integer operations or floating-point operations, as well as ALU or multiply/divide operations. The higher the latency of the corresponding functional unit, the higher the weight of the instruction. By minimizing the previous metric, we are weighting duty-cycle usage and functional unit performance. This can be easily done by sorting the ξ_j values at each scheduling period according to an ascending order.

4. EXPERIMENTAL RESULTS

We employed the proposed analysis methodology to assess the impact of different allocation strategies on the system-wide reliability of the reference single-core processor. In this section we will focus on the impact of the CS allocation strategy on different reliability-oriented metrics, when compared to classical RR or RS policies.

We selected several programs from freely available benchmark suites: WCET [7], SPLASH2 [20], MiBench [8], and FBench [21]. Each program in the different benchmark suite has been simulated using a modified version of the ESESC simulator, from the IACOMA group at the University of Illinois at Urbana Champaign [19]; during execution the instruction classes have been collected and statistically post-processed, to get the instructions breakdown

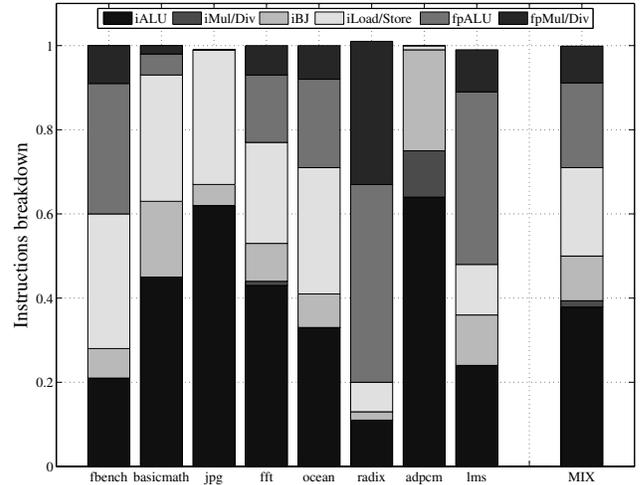


Figure 2: Instructions breakdown of the selected benchmarks, and total breakdown mix.

Table 2: Functional units characterization summary, for the simple five-stage MIPS processor.

Block	#total PMOS	#stressed PMOS (avg.)	
		absolute	relative
i-ALU	6653	3775	56.7%
fp-ALU	10061	6010	59.0%
fp-MULT	17083	9682	56.7%
fp-DIV	44873	22990	49.7%

for each benchmark of interest. Four different instruction classes have been selected for our experiments: integer ALU operations, floating-point ALU/multiplication/division instructions. We chose 8 benchmarks to run concurrently, such that each functional unit is used at a different mix. The instructions breakdown of the selected benchmarks is given in Figure 2. The reference processor is a simple five-stages pipeline MIPS processor, whose Verilog RTL design is freely available from OpenCores (opencores.org). The reference processor has one single integer ALU unit, one floating-point ALU, one floating point divider and a single floating-point multiplier. The RTL design has been synthesized using Cadence RTL Encounter Compiler, and a freely-available 45nm standard-cell library [12]. Verilog simulation has been conducted using NC-Verilog Simulator from Cadence as well. As specified in Section 3, the reference standard-cell library has been appropriately augmented. The results of the RTL simulation are given in Table 2: for each functional unit of interest, the number of PMOS transistors and the number of PMOS transistors that are stressed (as an average over the input vector configurations) is reported. Notice that the statistics reported in Table 2 are slightly different than the 50% assumption made in [17], providing a more precise and accurate evaluation methodology.

4.1 Impact of allocation strategy on reliability

The first analysis refers to the how long-term degradation benefits from the sole allocation strategy, while other technological and operational parameters remain constant. Figure 3 shows the trend of the V_{th} degradation process as a function of the time-wall considered (from 3 months up to 10 years), comparing RR, RS and CS scheduling for the floating-point multiplier block. The technology

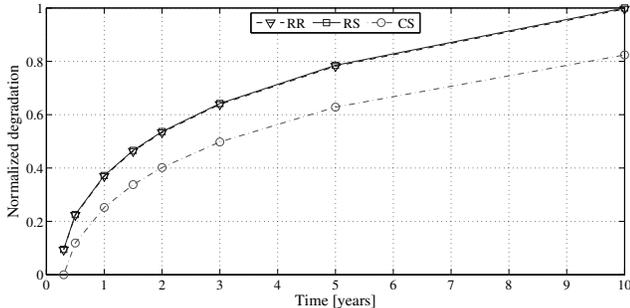


Figure 3: NBTI-induced V_{th} degradation as a function of reference lifetime.

node is set to 45nm, with supply voltage 1.1V and oxide thickness 9Å, as suggested in [10]. Each point on the trend curve is computed as an average across every stressed PMOS, according to the results obtained from the estimation flow presented in Section 3, and reported in Table 2. As the figure shows, Round-Robin and Random-Scheduling allocation strategies have nearly the same impact on the degradation process. This is an expected behavior, since in the RS scheduling process the duty-cycle in each functional unit tends very fast to the RR one due to the huge number of simulated cycles. On the contrary, the CS allocation strategy has an average benefit of at least 10% in the degradation, having a direct impact on the system life-time: the same reference ΔV_{th}^{thr} threshold is then reached later on, providing thus a greater life-time expectation. As an example, the degradation reached by RR scheduling after 1 year of operation is reached after 2 years with CS scheduling, with a net increase in system reliability. Lower degradation is seen throughout the entire typical lifetime periods for modern electronic products: results are well suited for consumer electronics, whose typical lifetime is in the order of 2 to 3 years, as well as for industrial/military electronics, in which long-term reliability is the primary objective. Nevertheless, this result is well suited for periodic or aperiodic tasks, as those we are addressing in this work; for continuous stress workload, however, benefits are expected to decrease over time since the duty-cycle usage typical of continuous stress would sooner reach the values of RR and RS. Without loss of generality, we set the operating temperature to 423 K. In the Arrhenius expression it is generally assumed that the temperature parameter is an average value [16], thus it simply causes a shift of the final estimation. More precise models should take into account the time-varying profile of temperature [24], also considering leakage. To analyze the impact of operating temperature on the benefits due to the different allocation strategies, we computed the degradation for increasing operating temperatures. With reference wall-time of 2 years and technology parameters as in the previous case, the operating temperature has been varied in the range [110, 210]°C; results have shown that the benefits of the proposed methodology have actually a temperature-independent behavior when compared to both RR or RS allocation strategies. Meanwhile, the degradation gets worse with increasing operating temperatures, as expected. As compared to RS scheduling, there is a reduction of degradation in the order of 10% in either case.

4.2 The impact of technology scaling

As supply voltage scales down, it is expected that degradation increases, since as V_{DD} scales the geometry of the transistor scales as well, such that the oxide thickness t_{ox} scales, too. The results in Table 3 show this trend. The values are taken according to increasing

Table 3: Impact of technology scaling: data is given as a function of supply voltage, considering a 2 years wall-time.

V_{DD}	Degradation		
	Round-Robin	Custom	Ratio (RR/CS)
0.8V	0.461V	0.42V	1.098
0.9V	0.458V	0.418V	1.096
1.0V	0.455V	0.413V	1.101
1.1V	0.44V	0.394V	1.117
1.2V	0.413V	0.37V	1.116

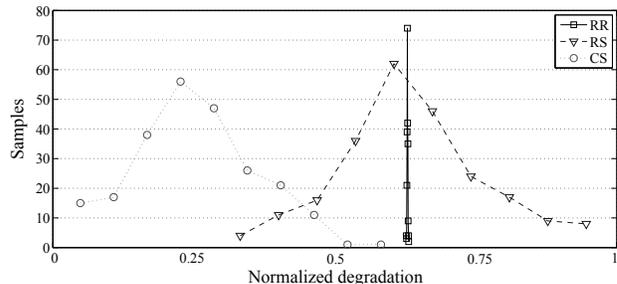


Figure 4: The impact of process variation and CS scheduling.

values of V_{DD} and t_{ox} , as frequently used in technology predictions [10]. Temperature is set to 383 K as usually done [16]. Technology values with prediction on 22 nm technology nodes have been derived from [10]. As in the previous cases, the NBTI-aware allocation strategy achieves benefits from a degradation view-point when compared to the classical RR or RS scheduling policies. In this perspective, the CS allocation is a suitable strategy to deal with NBTI degradation at architectural level, even in more aggressive technology scales. Comparing the degradation of CS and RS strategies at the same technology node, the degradation difference is in the order of 10%. Notice that similar results cannot be estimated with accuracy when considering high- κ dielectrics, since in those scenarios PBTI plays a relevant role as well [9], but in our work only NBTI is considered.

4.3 The impact of process variation

One important aspect of the proposed flow is the ability to capture the effects of process variation on a system-wide perspective. The RTL blocks characterization in Table 2 allows us to analyze the impact of V_{th} variation caused by process variability, and compute the run-time NBTI-induced degradation as a function of the allocation strategy. Process variation has negative impact on the technology parameters between devices on the same die or different die on the same wafer [1]. We are interested in particular on the first case, where different PMOS transistors can have a different (initial) threshold voltage. We assume the V_{th} distribution to be Gaussian, as shown in [1], and generate a random vector of V_{th} values according to a Gaussian distribution with mean $\mu = 0.18V$ and standard deviation $\sigma = 0.005$, such that the initial values are approximately in the range [0.17, 0.19]V. The shift toward lower values of the x axis in Figure 4 shows that the CS scheduling allows to have more transistors with lower (cumulative) degradation. However, it has also a greater variance, due to the increased variability of usage of the different transistors in the reference device. Indeed, RR has the higher mean degradation, but much lower variation. The mean value of the CS strategy is shifted by at least 30% toward lower values of degradation when compared to RR or RS.

This again proves the efficiency of the CS policy. Benefits are achieved from a performance view-point also; by applying the gate delay expression as a function of supply voltage and threshold voltage [14] with $\alpha = 1.3$ for 45nm technology node, the degradation follows similar shape as in Figure 4, with an average gain in gate delay of 30% relative value. This demonstrates that the proposed strategy can be efficiently employed while ensuring low performance degradation.

5. CONCLUSIONS

This paper presents a novel methodology to estimate the NBTI degradation at architecture-level, as a function of technology parameters, RTL design, and usage (i.e., duty-cycle and allocation strategies). A novel dynamic instruction allocation strategy has been proposed, and several experiments have been conducted to show the benefits of NBTI-aware allocation strategies to prolong the device lifetime. We compared this Custom Schedule strategy with classical Round-Robin and Random-Selection policies. Results have shown a reduction of degradation in the order of 10% to 20%, in different scenarios: at different operating temperature values, as technology scales down and, above all, as process variation is accounted for. Furthermore, results show that CS is well suited for both consumer electronics, gaining benefits in the range of 2 to 3 years lifetime, and long-term reliability of industrial/military electronics. The allocation strategy is meant to be employed in the dynamic instruction scheduling portion of the microprocessor pipeline. Nevertheless, this paper focuses on the execution stage of the pipeline focusing on the functional units rather than in processor control units, since it has been shown that the processor resources that are prone to higher failure rate are ALU and control blocks [6].

6. ACKNOWLEDGMENTS

This research work is partially supported by European Community Seventh Framework Programme (FP7/2007-2013), under agreements no. 248716 (2PARMA project www.2parma.eu) and Artemis no. 100230 (SMECY project www.smecy.eu).

7. REFERENCES

- [1] K. Agarwal and S. Nassif. Characterizing process variation in nanometer cmos. In *44th ACM/IEEE Design Automation Conference.*, pages 396–399, june 2007.
- [2] M. Alam, K. Kang, B. Paul, and K. Roy. Reliability- and process-variation aware design of vlsi circuits. In *14th International Symposium on the Physical and Failure Analysis of Integrated Circuits.*, pages 17–25, july 2007.
- [3] M. A. Alam and S. Mahapatra. A comprehensive model of PMOS NBTI degradation. *Microelectronics Reliability*, 45(1):71–81, 2005.
- [4] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula. Predictive modeling of the nbtI effect for reliable design. In *IEEE Custom Integrated Circuits Conference.*, pages 189–192, sept. 2006.
- [5] S. Borkar. Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation. *IEEE Micro*, 25(6):10–16, Nov. 2005.
- [6] T. Gupta, C. Bertolini, O. Heron, N. Ventroux, T. Zimmer, and F. Marc. Effects of various applications on relative lifetime of processor cores. In *IEEE International Integrated Reliability Workshop Final Report.*, pages 132–135, oct. 2009.
- [7] J. Gustafsson, A. Betts, A. Ermedahl, and B. Lisper. The Mälardalen WCET Benchmarks - Past, Present and Future. In *Proceedings of the 10th International Workshop on Worst-Case Execution Time Analysis*, July 2010.
- [8] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown. Mibench: A free, commercially representative embedded benchmark suite. In *Proceedings of the IEEE International Workshop on Workload Characterization.*, pages 3–14, Washington, DC, USA, 2001.
- [9] K. T. Lee, C. Y. Kang, O. S. Yoo, R. Choi, B. H. Lee, J. Lee, H.-D. Lee, and Y.-H. Jeong. PbtI-associated high-temperature hot carrier degradation of nmosfets with metal-gate/high- k dielectrics. *Electron Device Letters, IEEE*, 29(4):389–391, april 2008.
- [10] MASTAR - Model for Assessment of CMOS Technology and Roadmaps <http://www.itrs.net/models.html>.
- [11] J. W. McPherson. Reliability challenges for 45nm and beyond. In *Proceedings of the 43rd annual Design Automation Conference*, pages 176–181, New York, NY, USA, 2006.
- [12] NanGate Open Cell Library <http://www.si2.org/>.
- [13] Predictive Technology Model <http://ptm.asu.edu>.
- [14] T. Sakurai and A. Newton. Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas. *Solid-State Circuits, IEEE Journal of*, 25(2):584–594, apr 1990.
- [15] K. Saluja, S. Vijayakumar, W. Sootkaneung, and X. Yang. NbtI degradation: A problem or a scare? In *21st International Conference on VLSI Design*, pages 137–142, jan. 2008.
- [16] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers. The Case for Lifetime Reliability-Aware Microprocessors. In *International Symposium on Computer Architecture*, 2004.
- [17] J. Sun, A. Kodi, A. Louri, and J. Wang. NBTI aware workload balancing in multi-core systems. In *International Symposium on Quality of Electronic Design*, pages 833–838, march 2009.
- [18] J. Sun, R. Lysecky, K. Shankar, A. Kodi, A. Louri, and J. Wang. Workload capacity considering nbtI degradation in multi-core systems. In *15th Asia and South Pacific Design Automation Conference*, pages 450–455, jan. 2010.
- [19] Superscalar simulator <http://sourceforge.net/projects/sesc/>.
- [20] The modified SPLASH-2 benchmark suite <http://www.capsl.udel.edu/splash/>.
- [21] Trigonometry intense floating point benchmark <http://www.fourmilab.ch/fbench/fbench.html>.
- [22] R. Vattikonda, W. Wang, and Y. Cao. Modeling and minimization of pmos nbtI effect for robust nanometer design. In *43rd ACM/IEEE Design Automation Conference*, pages 1047–1052, 0-0 2006.
- [23] Y. Wang, X. Chen, W. Wang, V. Balakrishnan, Y. Cao, Y. Xie, and H. Yang. On the efficacy of input vector control to mitigate nbtI effects and leakage power. In *International Symposium on Quality of Electronic Design*, pages 19–26, march 2009.
- [24] B. Zhang and M. Orshansky. Modeling of NBTI-Induced PMOS Degradation under Arbitrary Dynamic Temperature Variation. In *9th International Symposium on Quality of Electronic Design*, pages 774–779. IEEE, Mar. 2008.