NBTI-Aware DVFS: A New Approach to Saving Energy and Increasing Processor Lifetime

Mehmet Basoglu   Michael Orshansky   Mattan Erez

The University of Texas at Austin
Electrical and Computer Engineering Department
Austin, TX 78712
{mbasoglu, orshansky, mattan.erez}@mail.utexas.edu

ABSTRACT

Scaling process technology necessitates the introduction of wide design-time guard bands that ensure lifetime reliability as circuits wear out over time. In this paper, we show how to utilize this knowledge of the guard band and a predictive model to absolutely improve processor power consumption and lifetime without impacting the processor performance against Negative Bias Temperature Instability (NBTI) degradation. For the first time, we evaluate the long-term potential and impact of NBTI-aware job-to-core mapping quantitatively and account for process variations in the system. Our approach saves up to 16% of the dynamic energy consumed and improve lifetime by two years.

Categories and Subject Descriptors

C.5.4 [**Computer System Implementation**]: VLSI Systems.

General Terms

Design, Economics, Management, Performance, Reliability.

Keywords

DVFS, Energy efficiency, NBTI, Process variation, Wearout.

1. INTRODUCTION

We describe, and quantitatively evaluate, a low-cost control framework that accounts for the impact of device wearout to reduce power consumption and increase processor lifetime. Our innovations are in combining an accurate dynamic wearout model with infrequent measurements for safely reducing supply voltage while at the same time mapping threads to maximize lifetime in the presence of process variation. Additionally, we propose, for the first time, a technique to model NBTI degradation with dynamic changes to temperature, voltage, and frequency. We rely on this model to study the impact of our techniques using data collected from real application execution combined with workload models. The results show up to two years of lifetime improvement while yielding up to 16% dynamic energy savings, all without compromising performance and without exposing variations.

As in prior work, our methods rely on the fact that while devices gradually wearout and switch more slowly, the processor must operate at the specified frequency for many years. Circuit designers use guard bands to ensure correct operation over the entire target lifetime. While some of these margins are inserted to prevent failures due to unpredictable short-term effects or to improve manufacturing yields, others are aimed to combat more predictable long-term wearout effects. The latter margins introduced at design time squander performance and energy efficiency during the life of the processor as the supplied voltage is higher than needed to run the system at its nominal frequency.

In this paper, we utilize the margins to reduce energy consumption on top of other techniques, such as DVFS, while maintaining the specified performance at all times. We use Negative Bias Temperature Instability (NBTI) as an exemplary wearout mechanism. NBTI causes the switching speed of p-MOS devices to gradually degrade. We show how to reduce power by choosing a lower operating voltage for every DVFS operating point, which we call the *NBTI voltage*. The NBTI voltage is based on estimating the remaining margin available and is strictly lower than the nominal DVFS voltage. To the best of our knowledge, this is a new contribution and has not been done before, including in [1] and [2], which we discuss in Section 2. Moreover, we go further and, for the first time, evaluate the long-term potential of wearout-aware job-to-core mapping by utilizing our detailed degradation model to guide the optimization. Additionally, using a model within the voltage control loop enables only very infrequent measurements of degradation. This significantly reduces the potential negative impact of continuously tracking available margins [3, 4, 5], which includes additional area and power for the sensors and possible increased wearout rate.

Another aspect of our research is that multi-core processors witness process variations, and different cores start with different threshold voltages. Consequently, some cores on the same chip start with shorter life expectancy than others. If no precaution is taken, the processor will die early as one core in the system fails before others. The work presented in [6] examines the impact of scheduling on processor lifetime based on statistical data and gives insight to the problem. We take a step forward by utilizing real-time degradation data and employ a technique that can help alleviate this problem through core-level DVFS control and OS-controlled workload mapping based on core status. If the OS is informed of the degradation status of each core, it could map threads so that *sturdy cores* (those with low threshold voltages) work more than the *frail cores* (those with high threshold voltages). This equalizes core lifetimes, and thereby extends overall processor life. Our experiments show that NBTI-aware mapping allows a processor to run up to two years beyond its targeted lifetime under heavy workloads and even longer under lighter loads, again, without impacting performance. Concurrently, by combining our techniques, the energy consumption of the system is reduced up to 16% over the target lifetime of the processor and even more for shorter periods.

Our work significantly advances the state of the art by combining the many ideas presented to mitigate the impact of gradual wearout; introducing a new NBTI model for dynamic voltage, temperature, and usage; accounting for process variations; proposing a low-cost control framework; and quantitatively analyzing the impact over years of operation using a methodology that combines degradation modeling, power estimation, workload modeling, and data collected from a real machine.

2. RELATED WORK

Prior work related to NBTI wearout in processors falls into three main categories: work similar to our own to mitigate NBTI; techniques to reduce NBTI effects; frameworks with fine-grained and speculative continuous adjustment of voltage and frequency.

Among the studies done in our field, the closest solutions to ours were proposed in [1] and [2] where workload management based on application temperature and chip-wide changes to supply and threshold voltages were recommended. While providing a comprehensive discussion of the impact of NBTI and methods to mitigate its detrimental effects, these studies do not take into account the existing DVFS mechanisms, and the former does not provide a quantitative evaluation. More importantly, the methods presented compromise performance to increase lifetime. If the approach of [1] is taken to the limit, one would see that the system life can be extended by multiple times of the original target lifetime by running the processor at very low voltages and frequencies; however, then, the performance of such a processor would be significantly reduced. The latter study, on the other hand, fails to account for the idle times in the system and evaluates a single circuit operating constantly, which would not be applicable to a chip multi-processor with process variations where the degradation should be based on the workload of the system.

The work presented in [6] is also similar to ours in terms of degradation-aware job-to-core mapping, but it is entirely statistical. It assumes a static "bathtub" curve to represent aging and only addresses the different rates of wearout due to unevenly-balanced workloads. We, on the other hand, incorporate the mapping within a model-based degradation-tracking framework that can calibrate itself. Moreover, we allow the OS to use any load-balancing scheduling algorithm and only modify the thread-to-core mapping.

Another technique suggested in [7] takes a different approach and proposes to reduce the rate of NBTI-induced wearout. Because NBTI occurs only when p-MOS devices are negatively stressed, the overall effect can be reduced if the time a transistor is stressed is limited by flipping the meaning of a logical zero and one to even out the degradation as well as allow time for the devices to recover. However, this technique is only effective if the duty cycle is unbalanced. Additionally, other work [8, 9, 10] question the magnitude and durability of recovery.

Approaches like continuous control [11, 12, 13] have also been recommended to eliminate all timing margins in a system. Despite the qualitative appeal of these ideas, their practical applications have often resulted in significant area and performance overheads [14]. Applying these techniques requires monitoring all potential critical paths in the processor [15]. Unfortunately, industrial designs balance all paths as best as possible, which results in large overheads for continuous monitoring and could outweigh the benefits of dynamic margin reduction. In addition, because the control is speculative, timing errors may occur and require corrections, which again increase overhead.

In summary, the work presented in this paper complements prior approaches and advances the state of the art. It can be used in conjunction with other mechanisms to reduce or tolerate wearout and offers a lower-cost approach to controlling and tracking margins than prior techniques. Our work is also differentiated by our focus on ensuring predictable performance levels, which match the processor specifications, throughout the entire lifetime while reducing energy consumption, by developing a degradation-aware mapping technique, and providing a quantitative evaluation of energy and lifetime benefits.

3. A MODEL FOR NBTI DEGRADATION

NBTI reaction-diffusion (R-D) models developed so far predict the degradation over time under fixed stress conditions. These models have been extended in [16, 17] to include dynamic temperature variations. However, we are not aware of any prior work that models the impact of NBTI under changing stress voltages, which is needed to estimate the degradation for the different DVFS voltages the processor may select. We improve the model of [16, 17] by including the impact of the changes in the stress voltage (V_{dd}). This enhancement enables our low-cost NBTI-Aware DVFS framework as well as the quantitative evaluation presented later in this paper.

3.1 Static Voltage and Temperature Stress

Basic NBTI models cover the impacts of stress voltage and temperature; however, these crucial parameters are considered to be fixed in these models during the full operation time. The shift in the threshold voltage due to NBTI is estimated using a formula of the following or similar form [16, 17]:

$$|\Delta V_{th}(t)| \propto C_0 \left[\left(e^{-\left(\frac{E_a}{k_B}\right)\left(\frac{1}{T}\right)} \right) (k_{FR}(V))^{(2/3)} t \right]^{\frac{1}{6}} \tag{1}$$

$$k_{FR}(V) \propto \left(\frac{V - V_{th0}}{t_{ox}} \right) e^{\left(\frac{V - V_{th0}}{t_{ox}} \right)\left(\frac{1}{E_0} \right)} \tag{2}$$

These equations state that the shift in the threshold has a monotonically-increasing relation to time, which will simplify our solution to the NBTI problem under dynamic stress conditions.

3.2 Dynamic Voltage and Temperature Stress

Changes in temperature and voltage have a significant impact on the rate of NBTI degradation. R-D models that do not capture these effects are not sufficient to predict the impact of NBTI on a processor running realistic workloads. Thus, we need to utilize a model that tracks the behavior of NBTI under dynamic stress. The model presented in [16, 17] gives a solution to the problem of dynamic temperature, and it does not require that the entire temperature history be known to predict the degradation in the future. Furthermore, the techniques used for estimating NBTI under temperature variations can also be applied to the changes in stress voltage, which are necessary to track degradation in a processor as the voltage varies under DVFS.

Given the monotonic threshold shift in the absence of recovery, we follow the conceptual idea presented in [16, 17], which represents the history of degradation using an equivalent stress time. We use equations (1) and (2) to estimate the degradation as long as the voltage and temperature are constant. A change in these parameters, however, cannot be simply entered into these equations because they would lead to a discontinuity in the threshold voltage, which is clearly wrong. Instead, we need to replace the current time t with an equivalent time t' that would result in the current degradation under the new stress conditions by solving equation (3) and then update it to $t' + \Delta t$.

$$\Delta V_T(t', V_2, T_2) = \Delta V_T(t, V_1, T_1) \tag{3}$$

Our solution in equation (3) essentially results in time shifts, which are easy to visualize. To replicate the impact of varying
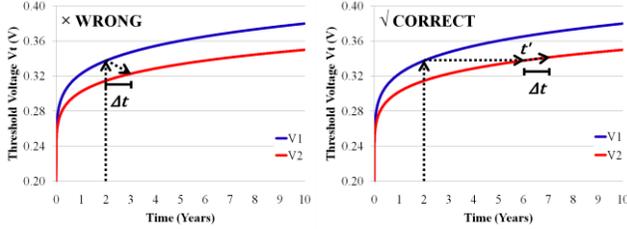
Figure 1. Threshold voltage is found for the first given DVFS voltage V1 and then mapped onto the curve for the second DVFS voltage V2 as shown on the right. Then, terminal V_T is read at $t' + \Delta t$.

Figure 2. The baseline CPU (unshaded) and the modifications (shaded).

stress on the processor due to different DVFS voltages, we follow the approach shown in Figure 1. For example, assume a core on a processor is run at voltage V1 for one period of time and V2 for the next period of time. The V_T of the core is found after the first interval using the degradation curve for V1, and this point is then located on the V2 degradation curve. Next, the core would run for another interval at this voltage, and the terminal V_T is read from the V_T curve for V2 at the end of this time period. If the core is powered down, our model does not change V_T. This is pessimistic because some recovery would occur, but there is no consensus in the community regarding the extent of recovery. In fact, some studies predict that recovery is very temporary and the device reaches to the highest point of degradation under the previous stress phase very rapidly when it is subjected to stress a second time [8, 10]. Nevertheless, we will present an inherent solution to the question of recovery later in this paper (Sections 4.2 and 6).

3.3 Model Parameters
Our new NBTI model, incorporating dynamic voltage stress into prior work, uses similar parameters to those presented in literature. First, the model is programmed with technology specific parameters such as the maximum strength of the electric field at the gate, thickness of gate oxide, activation energy, and threshold voltage. Also the model is provided with nominal conditions such as maximum DVFS voltage and expected average temperature. This data allows the model to predict the degradation in the system as the operating voltage and temperature varies.

4. NBTI-AWARE DVFS FRAMEWORK
To explain our framework, we use an example of a processor that can be built in today's fabrication technology, but our framework is applicable to future generations. We evaluate an 8-core processor in 45nm technology, which allows us to use real data on NBTI degradation [18]. While our framework is suitable for a single voltage regulator with reduced benefits, we choose to have per-core voltage and frequency regulators as suggested in [19].

NBTI-Aware DVFS requires three main modifications to current processors with insignificant area overhead (Figure 2). The power management unit [20] is augmented to track available margins based on the NBTI model and to determine the NBTI voltage that should be used based on the voltage ID (VID) supplied by DVFS (Section 4.1). Tracking degradation requires information on the stress voltage, which is known, as well as the core temperature, which requires temperature sensors. We also require a mechanism to perform periodic calibration in order to improve margin tracking, because the NBTI model is designed to be pessimistic to guarantee safe and correct execution (Section 4.2).

4.1 NBTI Voltage Control
The NBTI voltage regulation unit is a component placed in between the processor and the DVFS voltage regulator. This unit receives the VID signal from each core, which is generated based on the needs of the processor depending on the workload. Normally, these signals are directly sent to the DVFS voltage regulator. With NBTI-Aware DVFS, however, the power management unit intercepts this request of the processor. Then, based on a conservative estimate of the available timing margin in each core, this unit selects an appropriate VID corresponding to an NBTI voltage that is strictly smaller than or equal to the originally-requested DVFS voltage, which enables significant power savings. Figures 3a and 3b visualize the technique. They show how the nominal voltage wastes energy by starting with a wide margin, which decreases as the circuits wear out. Using NBTI-Aware DVFS, however, the margin can be kept roughly constant as the voltage is increased over time to always meet timing. Note that the NBTI model is conservative and will not drop the margin below a safe point. Due to the presence of different threshold voltages on the chip as a result of process variations and distinct utilization and temperature histories, the timing margin in each core is different. Thus, each core demands a distinct supply voltage in order to optimally meet the critical timing requirements. Furthermore, as the degradation progresses during the lifetime of the processor, our system needs to adjust itself to compensate for the decaying margins in each core.
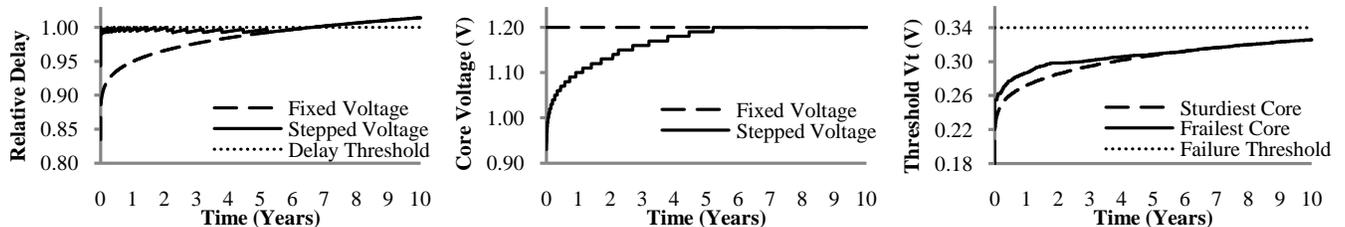

Figure 3. (a) Dynamic adjustment of the timing margin of a processor core as degradation progresses. (b) Dynamic adjustment of the voltage of a processor core as degradation progresses. (c) Convergence of degrading threshold voltages in a multi-core processor under lifetime maximization thread-to-core mapping.
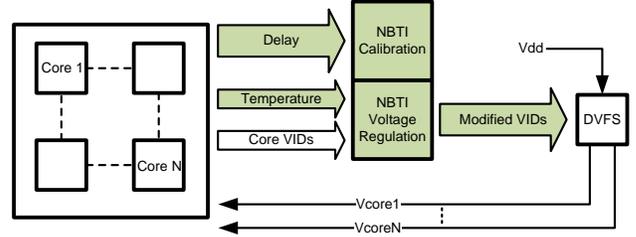
Maintaining the status of each core requires additional hardware and computation cost; however, we predict this overhead to be small enough that it is feasible to implement. Temperature sensors are already being placed throughout the chip, which we will also utilize to input parameters into our NBTI degradation model. The sample rate can be quite low as temperature varies relatively slowly, and we only need to update our model when the DVFS voltage is changed. We also require the NBTI voltage regulation unit to solve equation (3) every time a core operating condition is changed. This calculation is simple and can be done in roughly 24 floating point operations, again, only when a new VID is required.

4.2 NBTI Calibration

During normal operation, the NBTI voltage regulation unit tracks degradation and calculates the NBTI voltage based on its conservative estimate of the available margin. Because the model is conservative and cumulative, it requires periodic calibration to keep its estimate close to the actual degradation. NBTI degradation is a gradual process that builds up over time, and therefore, calibration can be performed with a slow period, which we estimate at several days. Every few days, the remaining margin of each core is estimated using direct measurement, and the measurement is used to update the model.

Calibration requires a method of directly measuring and estimating degradation or remaining margins. One approach, suggested in [3, 4, 5], places multiple delay sensors on the chip. The sensors are only turned on during calibration and thus do not consume much power and do not increase overall degradation, which might happen if they were kept on constantly as in continuous monitoring. While this hardware approach can estimate degradation very quickly and does not reduce performance, it has a drawback beyond the area overhead. The sensors need to be placed near the critical paths, so the critical paths must be identified at design time [15], and should be limited in number if the results are to be fully accurate.

An alternative to the hardware approach is to periodically run software tests while varying the supply voltage to gauge remaining margin and provide extra information to the degradation model. Each calibration routine will run pre-designed tests targeted to stress a large number of critical paths [21]. The test will be repeated for a range of voltages (all lower than nominal) at the nominal frequency to estimate the margin. Even if each calibration routine requires several seconds of execution, the impact on availability will be low because calibration occurs only once every several days on each core. In addition, calibration is a means of improving the power efficiency of the processor and is not required for correctness because the NBTI model is inherently pessimistic. Therefore, the OS has flexibility in scheduling calibration during the idle periods of a core.

A hybrid approach can also be used, in which hardware sensors provide more frequent feedback to the model and extensive software calibration is performed infrequently. The hardware sensors improve the energy saving potential because they provide extra information to the model and allow it to be less conservative on its margin estimate. The software tests are used to overcome the limitation of the hardware sensing approach with respect to critical paths. In the hybrid approach, software calibration can be very infrequent, and thus more extensive. The more extensive the calibration is the better the system can identify critical paths, and again reduce the degree of conservatism.

4.3 Variations & Wearout-Aware Mapping

Since we are looking at a multi-core chip, process variation across the processor is naturally expected and may cause the progress of degradation in each core be distinct. This necessitates the monitoring of degradation at different parts on the die since different cores will have different initial threshold voltages (V_T's) and will degrade at different rates. If the OS is informed of the degradation status of each core, it can make better decisions on job-to-core mapping and increase overall processor lifetime. In particular, we evaluate three task mapping schemes to study the implications of process variations: *lifetime minimization* (LT_{Min}), which results in the worst-case power and lifetime for a given workload; *random*, which is the expected case with conventional mapping; and *lifetime maximization* (LT_{Max}), which uses a greedy approach that tries to maximize overall lifetime.

LT_{Min} scheme maps most of the workload to the frailer cores in the system and thus causes the frailest core in the system to die at the earliest possible time under the given workload. This is an unrealistic mapping algorithm that we use as a base case for fair comparisons, because it provides a lower bound on lifetime. Note that LT_{Min} still benefits from NBTI-Aware DVFS. The second case we consider is random mapping, which does not take any action against degradation of the cores and distributes the work among the available cores randomly, as commercial operating systems do today. Finally, our third mapping algorithm (LT_{Max}) attempts to maximize lifetime and energy savings. It maps most of the work to the sturdier cores, and only uses frail cores when required to match the performance of the baseline processor. By saving the frail cores, the expected lifetime goes up, because all cores approach an equal level of degradation, which in return avoids the problem of frail cores dying prematurely (Figure 3c).

5. EVALUATION & RESULTS

Our evaluation is based on a wearout simulator that applies the new NBTI degradation model described in Section 3 to a multi-core chip. The simulator outputs the degradation state of each core and tracks the supply voltage at 1-second intervals. Furthermore, it reports the lifetime of each core, as well as the energy savings compared to a baseline processor with DVFS, but no NBTI-aware voltage regulation. We present more details on the simulated system and workloads below and then quantitatively evaluate the improvements to lifetime and energy.

5.1 System Model

As described in Section 4, we evaluate an 8-core processor at 45nm technology in this paper. We base this processor on the specifications of the Intel Q6600 quad-core processor, and scale the number of cores to 8 and the DVFS voltage range to 1.06 - 1.20V (from the 1.17 - 1.33V observed on the 65nm Q6600). In addition, we assume each core has its own voltage regulator and that the system supports the NBTI-Aware DVFS framework of Section 4. We follow the process parameters of the ITRS roadmap [22] and use an initial nominal V_T of 0.20V. To account of process variations, we vary each core's initial V_T by up to ±10%, based on the analysis of [23]. We also choose a, somewhat arbitrary, end-of-life V_T of 0.34V that corresponds to about 6 years of life for a fully-loaded processor core with nominal V_T.

We are evaluating the processor over its entire multi-year lifetime, and therefore could not use a traditional benchmark suite directly. Instead, we used a methodology that combines a statistical

workload model with data measured on an Intel Q6600 processor running the SPEC CPU2006 suite [24]. We ran the benchmarks repeatedly on a system with a Q6600 running Microsoft Windows Vista and collected traces of core temperatures and the DVFS voltages selected by the system at a granularity of 1 second for each benchmark run (benchmarks were run in isolation). We then used this information to generate a trace of tasks that the system simulator processed. We rely on the Lublin workload model [25] to generate task arrival times, durations, and the number of required cores, as well as a task type, which we use to differentiate between SPEC INT and SPEC FP benchmarks, and for each task, randomly select a SPEC benchmark and provide its voltage and temperature trace for all the cores assigned to the task.

To mimic the OS scheduling policies described earlier, the simulator performs a scheduling decision at a granularity of 1 second. During each 1-second interval, we determine how many cores are required to handle the tasks that will be run on the processor based on our workload model. Then, we select that many sturdy cores in our system to run these tasks. Next, depending on voltages requested by the tasks, which reflect the utilization of each processor core, we select the appropriate NBTI voltage for each "ON" core that can handle the workload with no degradation in performance, which are selected from the set of 0.76 - 1.20V at 0.01V increments by our model. Since the number of cores and the DVFS voltage for each core is determined by the workload itself, our approach does not degrade the performance by any means. For example, if all cores are needed for any given day with 100% utilization in each core, our model allows the entire processor to be used to the full extent. On the other hand, if only one core is needed, then only the sturdiest core in the system is utilized when we run in lifetime maximization mode. Similarly, we also simulate the cases where the frailer cores are turned on first for our lifetime minimization mapping, or we select any core regardless of their degradation for random mapping.

5.2 Lifetime Improvement

Our first set of experiments is designed to measure the impact of NBTI and NBTI-aware job-to-core mapping on processor lifetime. In particular, we look at different cases of process variations across cores, which result in distinct lifetime improvements. With larger variations, more frail cores exist, which results in potentially very short lifetimes if NBTI effects are not considered. In our experiments, we allowed cores to vary by ±10, $\pm5\%$, or 0% with respect to the nominal V_T. We refer to the variation of each core using the following notation:
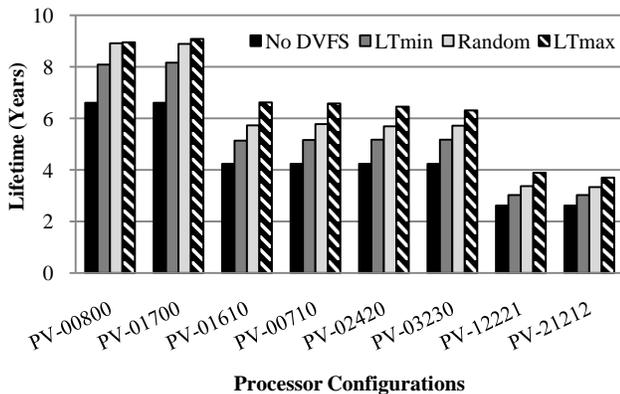
PV-ABCDE where PV stands for process variation, and A, B, C, D, and E represents the number of cores with −10, −5, 0, +5, and +10% process variations on the chip. Thus, a PV-00800 processor is an ideal 8-core processor with no variations from the fabrication point of view, which we defined as our base case.

We first look at how our base case PV-00800 behaves while running tasks from our combined SPEC/Lublin workload, and see that this processor can run for 2,954, 3,255, and 3,269 days under LT_{Min}, random, and LT_{Max} schemes respectively (Figure 4). As LT_{Min} scheme indicates the worst that can happen to a processor under the given workload, we report our improvements over this case. Thus, for the PV-00800, we see an approximate improvement of a year under the random and LT_{Max} mapping; however, there is no significant advantage to utilizing LT_{Max} over random mapping in this case.

We consider seven more cases for our lifetime study with increasing degrees of process variation, which clarify the benefits of our LT_{Max} mapping. Our results show that as the amount of process variation in the system increases, the gap between LT_{Max} and random schemes widens in general as a percentage of lifetime savings. However, on the absolute time scale, slight process variations (those within $\pm5\%$) yield the most savings of up to two years over the worst and a year over the average cases, under the heavy workload we use. PV-01700 has the longest lifetime among all configurations since the process variation present in this configuration leads to a sturdier-than-average processor overall. PV-21212 has two very frail cores, which cannot be fully overcome by LT_{Max}. The workload we use is heavy, and one of the frail cores fails earlier than the rest. Essentially, the initial margin was too small or the workload too heavy for all cores to converge to equal lifetime in this configuration.

5.3 Energy Savings

To estimate the energy savings of the system on top of those achieved by DVFS, we look at the NBTI voltage selected by our framework with the savings estimated for each core independently. Every second, the NBTI voltage selected for each core is compared against that normally chosen by DVFS alone. The instantaneous power savings are then computed by equation (4) for each utilized core and averaged over the sum of the times each core in the system is "ON" using equation (5). The energy savings is then determined by multiplying the power savings and the observed time period together in equation (6).
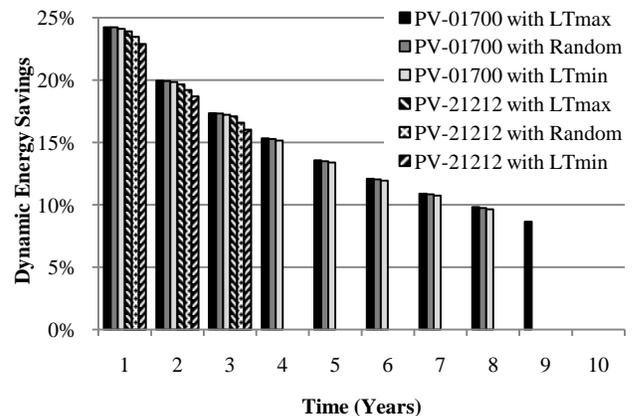

Figure 4. Lifetime improvements under various processor configurations.

Figure 5. Dynamic energy savings over time for PV-01700 and PV-21212 processors.

$$Instant.\ power\ savings(t) = 1 - \frac{(NBTI\ voltage\ (t))^2}{(DVFS\ voltage\ (t))^2} \quad (4)$$

$$\overline{Power\ savings(t)} = \frac{\sum_{k=1}^{\#Cores} \sum_{i=0}^{t} Instant\ .\ power\ savings\ (i)_k}{\sum_{k=1}^{\#Cores} ON\ time_{\ k}} \quad (5)$$

$$Energy\ savings(t) = \overline{Power\ savings(t)} \times t \quad (6)$$

Our results show that energy savings do not have a significant correlation to the mapping scheme chosen but rather to the lifetime of the system. As the processor survives longer, the energy savings drop due to the decaying guard bands in the system. As a result, NBTI-Aware DVFS can only take advantage of smaller and smaller margins over time. Despite LT_{Max} mapping always performing the best in terms of power savings, the numbers fall within a percent of each other.

Since the energy savings depend on the processor lifetime, we present the results of the longest and shortest running processors from our list of cases with different process variations (Figure 5). It should be noted that since PV-01700 has larger margins in the system than evenly-balanced cases of process variation, it has the highest power savings. Energy savings begin at 24.25% for the first year and keep dropping over time. As processors fail under different mapping schemes, we remove their results from the chart. Thus, we see that there are no results for PV-21212 after the third year. Similarly, PV-01700 survives past the ninth year only under LT_{Max} mapping and achieves energy savings of 8.66%. No configuration can continue its operation into the tenth year.

6. CONCLUSION & FUTURE WORK

In this paper, we presented an advanced NBTI model that allows us to predict NBTI degradation on chip with very little dependence to degradation measurements. We then used this model to build our low-cost NBTI-Aware DVFS framework, which reduces the energy consumption of the processor over its target lifetime by up to 16%. Moreover, we devised a simple lifetime maximization (LT_{Max}) mapping scheme that proactively balances the workload to get the most lifetime out of a processor by trying to equalize the degradation in all cores and yields up to two years of improvement. All of these techniques come with no negative impact on system performance, which differentiates our work from others mentioned earlier.

Our results also show that energy savings with NBTI-Aware DVFS have a significant correlation to target lifetime. As this target increases, the amount of the savings drops over time. Another conclusion we draw from our work is that process variations significantly reduce lifetime. If no action is taken, one of the cores can die prematurely and might render the rest of the processor useless. Therefore, proactive measures, such as LT_{Max} mapping, are required to combat process variations rather than placing wide guard bands at design time.

We believe that our work can be extended to include the recovery affects of NBTI, which would result in even longer lifetimes and more energy savings. At this point, we have left the corrections due to absence of recovery in our model to the calibration phase of our system as there is no consensus in the community on recovery. Another aspect of future work is to evaluate and improve on the current options for calibration, which will be required before we implement our technique in practice. Finally, while we do not provide quantitative results for lighter loads in this paper, we expect lifetime and energy savings to increase even further under less loaded scenarios.

7. REFERENCES

[1]   A. Tiwari and J. Torrellas, "Facelift: Hiding and Slowing Down Aging in Multicores," in *MICRO-41*.

[2]   L. Zhang and R. P. Dick, "Scheduled Voltage Scaling for Increasing Lifetime in the Presence of NBTI," in *ASP-DAC 2009*.

[3]   J. Keane, T. Kim, and C. H. Kim, "An on-chip NBTI sensor for measuring PMOS threshold voltage degradation," in *ISLPED '07*.

[4]   Z. Qi and M. R. Stan, "NBTI resilient circuits using adaptive body biasing," in *GLSVLSI '08*.

[5]   K. Stawiasz, K. A. Jenkins, P.-F. Lu, "On-Chip circuit for monitoring frequency degradation due to NBTI," in *IRPS 2008*.

[6]   F. Paterna *et al.*, "Adaptive Idleness Distribution for Non-Uniform Aging Tolerance in MultiProcessor Systems-on-Chip," in *DATE '09*.

[7]   J. Abella, X. Vera, and A. Gonzalez, "Penelope: The NBTI-Aware Processor," in *MICRO-40*.

[8]   M. A. Alam and S. Mahapatra, "A comprehensive model of PMOS NBTI degradation," in *Microelectronics Reliability*, vol. 45, no. 1. Jan. 2005.

[9]   S. Bhardwaj *et al.*, "Predictive Modeling of the NBTI Effect for Reliable Design," in *CICC '06*.

[10] M. Ershov *et al.*, "Degradation dynamics, recovery, and characterization of negative bias temperature instability," in *Microelectronics Reliability*, vol. 45, no. 1. Jan. 2005.

[11] K. Constantinides *et al.*, "BulletProof: a defect-tolerant CMP switch architecture," in *HPCA '06*.

[12] D. Ernst *et al.*, "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," in *MICRO-36*.

[13] S. Shyam *et al.*, "Ultra Low-Cost Defect Protection for Microprocessor Pipelines," in *ASPLOS 2006*.

[14] J. Sartori and R. Kumar, "Characterizing the Voltage Scaling Limitations of Razor-based Designs," Coordinated Science Laboratory, The University of Illinois at Urbana-Champaign, Champaign, IL, Tech. Rep. 2009.

[15] S. Ghosh, S. Bhunia, and K. Roy, "CRISTA: A New Paradigm for Low-Power, Variation-Tolerant, and Adaptive Circuit Synthesis Using Critical Path Isolation," in *TCAD*, vol. 26, no. 11. Nov. 2007.

[16] B. Zhang, and M. Orshansky, "Modeling of NBTI-Induced PMOS Degradation under Arbitrary Dynamic Temperature Variation," in *ISQED 2008*.

[17] B. Zhang, "Online Circuit Reliability Monitoring," in *GLSVLSI '09*. ACM, New York, NY, 221-226.

[18] J. Hicks *et al.*, "45nm Transistor Reliability," in *Intel Technology Journal*, vol. 12, no. 2. Jun. 2007.

[19] W. Kim, M. S. Gupta, G.-Y. Wei, D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," in *HPCA '08*.

[20] Intel Corporation, "Intel Core i7-900 Desktop Processor Extreme Edition Series and Intel Core i7-900 Desktop Processor Series Datasheet," vol. 1, rev. 3. Oct. 2009.

[21] I. Wagner and V. Bertacco, "Reversi: Post-silicon validation system for modern micro-processors," in *ICCD 2008*.

[22] ITRS, "Overall Roadmap Technology Characteristics." 2008.

[23] M. Royd *et al.*, "System power management support in the IBM POWER6 microprocessor," in *IBM Journal of Research & Development*, vol. 51, no. 6. Nov. 2007.

[24] SPEC, "SPEC CPU2006." Jun. 2008.

[25] U. Lublin and D. G. Feitelson, "The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs," in *Journal of Parallel & Distributed Computing*, vol. 63, no. 11. Nov. 2003.