Lifetime Reliability Enhancement of Microprocessors: Mitigating the Impact of Negative Bias Temperature Instability

HYEJEONG HONG, JAEIL LIM, HYUNYUL LIM, and SUNGHO KANG, Yonsei University

Ensuring lifetime reliability of microprocessors has become more critical. Continuous scaling and increasing temperatures due to growing power density are threatening lifetime reliability. Negative bias temperature instability (NBTI) has been known for decades, but its impact has been insignificant compared to other factors. Aggressive scaling, however, makes NBTI the most serious threat to chip lifetime reliability in today's and future process technologies. The delay of microprocessors gradually increases as time goes by, due to stress and recovery phases. The delay eventually becomes higher than the value required to meet design constraints, which results in failed systems. In this article, the mechanism of NBTI and its effects on lifetime reliability are presented, then various techniques to mitigate NBTI degradation on microprocessors are introduced. The mitigation can be addressed at either the circuit level or architectural level. Circuit-level techniques include design-time techniques such as transistor sizing and NBTI-aware synthesis. Forward body biasing, and adaptive voltage scaling are adaptive techniques that can mitigate NBTI degradation at the circuit level by controlling the threshold voltage or supply voltage to hide the lengthened delay caused by NBTI degradation. Reliability has been regarded as something to be addressed by chip manufacturers. However, there are recent attempts to bring lifetime reliability problems to the architectural level. Architectural techniques can reduce the cost added by circuit-level techniques, which are based on the worst-case degradation estimation. Traditional low-power and thermal management techniques can be successfully extended to deal with reliability problems since aging is dependent on power consumption and temperature. Self-repair is another option to enhance the lifetime of microprocessors using either core-level or lower-level redundancy. With a growing thermal crisis and constant scaling, lifetime reliability requires more intensive research in conjunction with other design issues.

Categories and Subject Descriptors: C.5.4 [**Computer System Implementation**]: VLSI Systems; D.4.1 [**Operating Systems**]: Process Management–Scheduling

General Terms: Design, Reliability

Additional Key Words and Phrases: Negative bias temperature instability, microprocessor, performance and reliability

1. INTRODUCTION

Ensuring lifetime reliability of microprocessors has become an important issue. Continuous technology scaling has enabled significant improvement of microprocessor
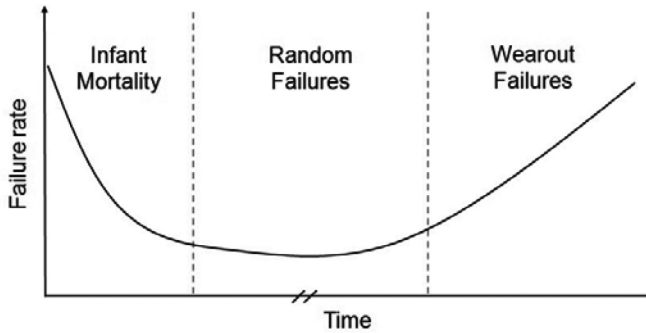
9

Fig. 1.   The classical bathtub curve.

performance but, coupled with increasing power densities, it has become a major threat to the lifetime reliability of microprocessors [Mostafa et al. 2012]. Lifetime of a microprocessor is defined as a period of time before the microprocessor has a permanent fault or cannot meet the design constraints due to wear out. Microprocessors should not fail in the middle of a normal lifetime that the designer intended; in other words, they should guarantee lifetime reliability.

Errors could be assorted into two types: soft and hard errors [Srinivasan et al. 2004]. Soft errors, also called transient faults or single-event upsets, are errors that occur while processors are active due to electrical noise or external radiation rather than to manufacturing defects. Transient faults may cause computation errors and corrupted data, but they are temporary and do not affect the lifetime of microprocessors.

Hard errors that are caused by silicon defects are usually permanent. They are classified into extrinsic failures and intrinsic failures. The main cause of extrinsic failures is process and manufacturing defects. Figure 1 is the classical bathtub curve that illustrates the failure rate over time. The failures according to each time stage are also specified. Extrinsic failures result in infant mortality with a decreasing rate over time as shown in Figure 1. Chips with extrinsic failures are screened out through burn-in tests, which is one of the major test processes that tests chips under stress with high temperatures and voltages to accelerate the extrinsic failures. Unscreened extrinsic failures can cause device failures very early, thereby extremely shortening the lifetime of the microprocessor. Thus, semiconductor manufacturers and chip companies have conducted extensive research on improving burn-in efficiency and reducing extrinsic failure rates. Currently, the major reason for lifetime failures is intrinsic failure. Intrinsic failures are wear-out failures that are generated by maneuvers within the stipulated conditions with an increasing rate over time. These failures largely hinge on the materials and are associated with process parameters, packaging, processor design, and so on. Causes of intrinsic failures are electromigration, stress migration, thermal cracking, and negative bias temperature instability (NBTI). NBTI has been known for several decades, but its impact was considered to be negligible. However, NBTI has become significant in today's typical SiO_2 operating fields because of aggressive process scaling. NBTI is now accepted as one of the dominant factors causing lifetime reliability problems on chips due to the increase of the threshold voltage of p-MOSFET transistors [Abadeer and Ellis 2003; Huard and Denais 2004].

In this article, we introduce recent techniques to enhance the lifetime reliability of microprocessors by mitigating NBTI degradation. Figure 2 illustrates the hierarchical structure of the rest of the article. The NBTI mechanism and its impact on lifetime reliability are presented in Section 2, prior to introducing the details of various techniques. The measurement and estimation of NBTI degradation are also presented. The

Mitigating the impact of NBTI

2. NBTI
   - 2.1. NBTI Mechanisms
   - 2.2. Lifetime Reliability Model ——— Increase of the Threshold Voltage and Delay
     - Mean Time to Failure (MTTF)
   - 2.3. Measurement and Monitoring ——— Extra Hardware
     - Software-Based
     - Hybrid
   - 2.4. PBTI

3. Circuit-Level Techniques
   - 3.1. Design-Time Techniques
     - 3.1.1. Gate/TR Sizing
     - 3.1.2. NBTI-Aware Synthesis ——— Input Vector Control (IVC)
       - Internal Node Control (INC)
   - 3.2. Adaptive Techniques
     - 3.2.1. Forward Body Biasing (FBB)
     - 3.2.2. Adaptive Voltage Scaling (AVS)
     - 3.2.3. Hybrid Techniques

4. Architectural Techniques
   - 4.1. Power Gating
   - 4.2. Self-Repair

5. System-Level Techniques
   - 5.1. Scheduling/Load Balancing ——— Instruction Scheduling on Functional Units
     - Task Scheduling on Multicore Processors
     - Workload Balancing on Multicore Processors
   - 5.2. Dynamic Voltage and Frequency Scaling
   - 5.3. Self-Repair ——— Utilizing Redundancy Cores on Multicore Processors
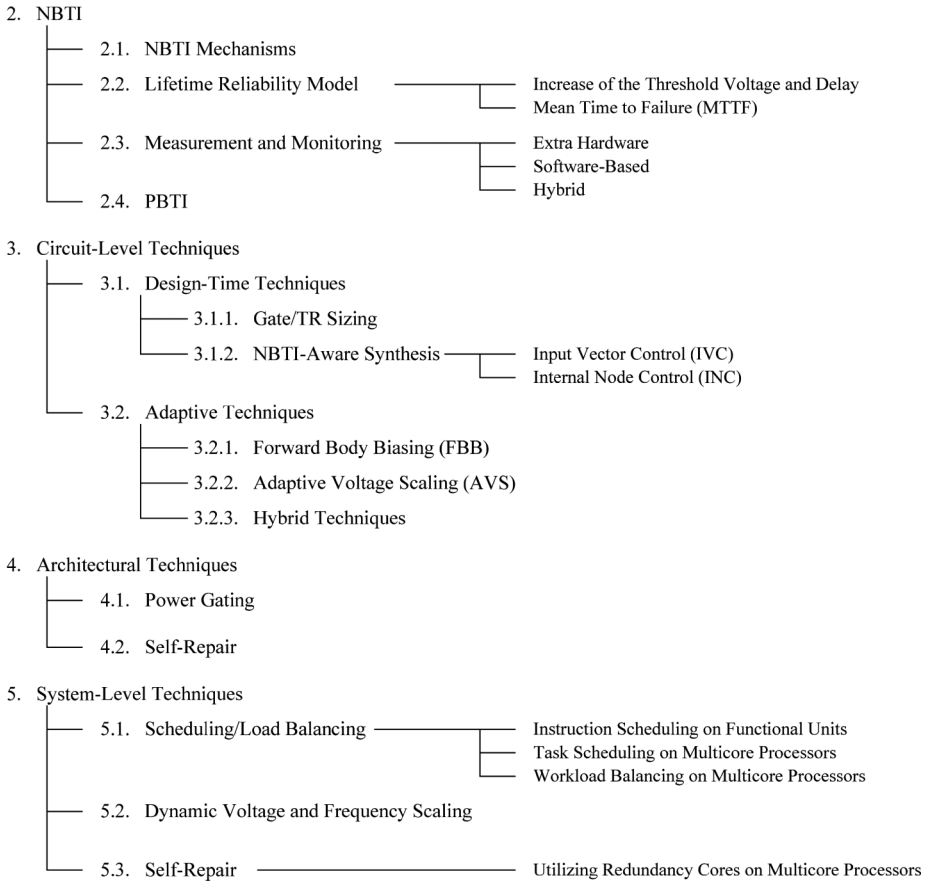
Fig. 2. Hierarchical structure of the article.

dual effect of NBTI, positive bias temperature instability (PBTI), is briefly introduced at the end of Section 2. The NBTI mitigation techniques can be classified according to level. The circuit-level techniques presented in Section 3 include design-time techniques such as transistor sizing and NBTI-aware synthesis. Other circuit-level solutions are adaptive techniques: forward body biasing (FBB) and adaptive voltage scaling (AVS), which control the threshold voltage or supply voltage to hide the lengthened delay caused by NBTI degradation. Traditionally, reliability problems are regarded as something to be addressed by chip manufacturers. However, there have been recent attempts to bring lifetime reliability problems to the higher design level. The architectural techniques in Section 4 and system-level techniques in Section 5 can reduce the cost added by circuit-level techniques, which are based on the worst-case degradation estimation. Traditional low-power and thermal management techniques can be successfully extended to deal with reliability problems since aging is dependent on power consumption and temperature. Self-repair is another option to enhance the lifetime of microprocessors using either core-level or lower-level redundancy, which is also presented in Sections 4 and 5. In Section 6, we summarize this work and offer some conclusions.
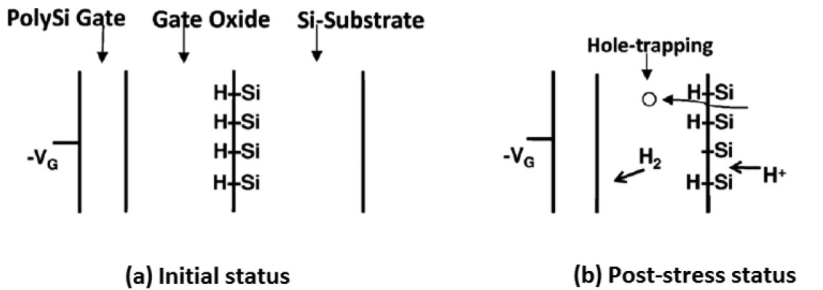
Fig. 3. NBTI degradation mechanism [Singh et al. 2012].

2. NBTI

Bias temperature instability (BTI) is a deterioration behavior that has been observed in MOS Field Effect Transistors (MOSFETs) since the late 1960s [Deal et al. 1967; Frohman-Bentchkowsky 1971]. Although a consensus has not been reached regarding the exact causes of the degradation, it is now generally understood that either at the Si/SiO_2 interface or in the oxide layer, positive charges are produced in a consistent gate voltage and increased temperature, leading to the performance reduction [Deal et al. 1967; Frohman-Bentchkowsky 1971]. This degradation has been regarded as insignificant for several decades, particularly when compared to the degradation caused by hot carrier injection (HCI) due to the excessive use of channel devices at that time.

The width of the gate oxide layer has been continuously decreased. To compensate for the caused performance loss, various nitridation processes are involved in the insertion of nitrogen atoms into the oxide layer. The nitridation step is to control the gate leakage current and to prevent the boron atoms from flowing through the oxide into the substrate. In addition, it is common that multigate devices are usually used rather than surface-channel devices in recent process technologies. This aims to improve performance and counter the short-channel effect, which has been exacerbated due to aggressive scaling. The introduction of the nitridation step and the utilization of surface-channel devices has resulted in a lot of research attributing an enhanced degradation of p-MOSFETs to negative bias and increased temperatures, named NBTI degradation [Uwasawa et al. 1995; Ogawa and Shiono 1995; La Rosa et al. 1997].

2.1. NBTI Mechanisms

As stated earlier, an agreed-upon account is yet to be provided for the precise nature of NBTI, but most research attributes it to two mechanisms. The first considers interface traps and oxide charge forming caused by negative gate bias at high temperatures. It involves the crack of Si–H bonding at the Si/SiO_2 interface because of electric field, temperature, and holes. The mechanism is illustrated in Figure 3; Figure 3(a) represents the initial status of a device and Figure 3(b) represents the device after undergoing the stress phase. This comes to dangling bonds or interface traps at the interface and a positive oxide charge [Huard et al. 2007]. The change of threshold voltage because of stress is lasting and not able to be recuperated even after the elimination of the stress. The second mechanism considers hole-trapping caused by the electric field in the gate oxide. The threshold voltage change is partially recovered after the removal of the negative bias. The reason for this is hole detrapping in the gate oxide [Huard et al. 2007].

As it undergoes repeated stress and recovery, the threshold voltage of a device is gradually increased. NBTI becomes a crucial challenge for reliability since the significant increment in threshold voltage can come to marginal operation [Paul et al. 2005;

Wang et al. 2007a]. NBTI also degrades the carrier mobility over time. It is important to exactly model the mobility degradation due to NBTI for NBTI-aware design. The modeling of mobility degradation has been studied in two previous works [Ayala et al. 2011; Chaudhary and Mahapatra 2013]. Additionally, NBTI degradation in p-MOSFET devices is known to generate a decrement in the static noise margin of SRAM cells, resulting in cell stability problems due to read operations [Kumar et al. 2006].

2.2. Lifetime Reliability Model

There have been many studies to model the lifetime reliability effect of NBTI degradation. Since there is no perfect consensus about NBTI mechanism, modeling has been challenging. During the measurement of the stress of the device, it is necessary for the measurement time to be small enough to evade covering the real threshold voltage movement by inadvertent recovery [Vattikonda et al. 2006]. This has been one of the biggest challenges to model NBTI. It is very hard to estimate the chip lifetime experiencing NBTI due to the recovery effect. The reason for this is that the chip characteristic alters due to stress immediately after it becomes likely to be recovered.

In order to cover static and dynamic NBTI degradation, a prediction model is introduced in the device level in Vattikonda et al. [2006], who show the accuracy of this device-level model using results from the experiments. The model is used to compute the amount of threshold voltage change caused by NBTI degradation over time using equations for stress and recovery phases:

Stress Phase:

$$\Delta V_{th} = \sqrt{K_v^2 (t - t_0)^{1/2} + \Delta V_{th0}^2 + \delta_v},$$ (1)

Recovery Phase:

$$\Delta V_{th} = (\Delta V_{th} - \delta_v)\left(1 - \sqrt{\eta(t - t_0)/t}\right),$$ (2)

where,

$$K_v = A t_{ox} \sqrt{C_{ox}(V_{gs} - V_{th})}\left(1 - \frac{V_{gs}}{V_{gs} - V_{th}}\right) exp\left(\frac{E_{ox}}{E_o}\right) exp\left(\frac{E_a}{kT}\right).$$ (3)

A description of the predictive model and technology specifications were addressed in detail in the original article [Vattikonda et al. 2006]. Using these equations, it is possible that the threshold voltage change of every p-MOSFET transistor is decided at each cycle. After simulation of the specified lifetime, the increase in delay of each transistor can be computed using the derived threshold voltage change based on the alpha-power law model [Sakurai and Newton 1990], which shows the relationship between the transistor delay and threshold voltage as:

$$Delay \propto \frac{C_L V_{dd}}{(V_{dd} - V_{th})^\alpha},$$ (4)

where C_L is the gate output load capacitance, V_{DD} represents the supply voltage, V_{th} is the gate threshold voltage, and α is the velocity saturation constant for short channel effects. The entire circuit delay can be obtained with the help of timing analysis methods using the elevated transistor delay. The lifetime of a circuit can be defined as the time point at which the overall delay of the circuit reaches the predefined maximum value that meets the performance constraints. Despite its accuracy, this cycle-by-cycle model for NBTI degradation may not be desirable due to the tremendous computation costs.

Today's microprocessors run at the order of GHz. A lifetime is normally several years, thus cycle-by-cycle lifetime simulation requires a large amount of computation. By scaling up the time unit of the simulation, this model can be widely utilized.

The NBTI model has been extensively studied to take various factors into account. Process variation can affect lifetime reliability of microprocessors since the initial threshold voltage can vary due to process variations. NBTI and process variation jointly affect lifetime reliability. A statistical NBTI degradation model was introduced by Lu et al. [2009] and Siddiqua et al. [2011], which considers process variation. Temperature is another critical factor that affects lifetime reliability, but traditional NBTI models have not considered temperature variations. Hamdioui [2010] showed how much temperature variations affect NBTI degradation and proposed a new NBTI model that considers temperature variations. Some works proposed a unified degradation model of NBTI and other reliability issues. Wang et al. [2011] proposed a unified model of NBTI and HCI, proving the accuracy and efficiency of the model.

Lifetime can be also expressed using mean time to failure (MTTF). MTTF by NBTI can be modeled as:

$$MTTF_{NBTI} = \left\{ \left[\ln\left(\frac{A}{1 + 2e^{B/kT}}\right) - \ln\left(\frac{A}{1 + 2e^{B/kT}} - C\right) \right] \times \frac{T}{e^{-D/kT}} \right\}^{1/\beta}, \quad (5)$$

where A, B, C, D, and β represent appropriate parameters, and k represents Boltzmann's constant. This lifetime reliability model was built on experiments done at IBM [Zafar et al. 2004].

2.3. Measurement and Monitoring

Measuring and monitoring degradation due to NBTI is the foundation to mitigating it. Measuring and monitoring can be categorized in two ways: one is the addition of extra hardware, such as sensors; the other is software-based technique. In the early stages of NBTI measurement, researchers used invasive probing methods that directly accessed the device-under-test (DUT) to monitor currents. Direct-current current–voltage (DCIV) is a method that can catch interface traps by monitoring the recombination current caused by interface traps [Neugroschel et al. 1995]. Using DCIV, the density of the interface trap in p-MOSFETs can be precisely checked while transistors are under stress. A number of previous works [Chen et al. 2002; Rangan et al. 2003; Huard et al. 2006; Denais et al. 2004; Aota et al. 2005; Shen et al. 2006; Ferńandez et al. 2006] have employed a method that can probe current directly.

Configurations based on ring oscillators were introduced in Kim et al. [2008], Keane et al. [2009], and Ketchen et al. [2007]. These configurations are composed of two ring oscillators. One is under intense stress and the other is not. The configuration in Kim et al. [2008] and Keane et al. [2009] measures the beat frequency, that is, the difference in the two oscillator frequencies. The threshold voltage shift of the ring oscillator exposed to elevated stress results in the decrease of frequency. Since these configurations generate digital signals, it is easy to gather and handle the output. A beat frequency detection method was developed to capture the effects of both DC and AC stress signals on NBTI aging [Kim et al. 2008]. They utilize fully digital differential measurements based on free-running ring oscillators, with minimal calibration and subpicosecond sensing resolution. It is necessary to have an off-chip analog bias to chart the altercation in beat frequency to move in the configuration in Ketchen et al. [2007]. Keane et al. [2010] also utilized an on-chip analog bias with the help of a DLL. The ring oscillators are substituted with delay lines. Using analog output is more difficult to gather and handle the data compared to using digital output. It is necessary to include a lot of delay stages in the ring oscillator–based configurations to achieve a

satisfactory sensitivity. Therefore, they are regarded to be inappropriate to use in large numbers as on-chip sensors due to the increased area.

Compact sensors for devices undergoing NBTI degradation are proposed in Singh et al. [2012]. The oxide degradation sensor is used to monitor the altercation of leakage under stress. Unlike the ring oscillator–based structures, this structure is small sized and low power. The concise sensors can be implemented in many ways in order to accumulate a large quantity of data.

An alternative to the hardware approach, which requires additional area overhead, is to periodically run software tests while varying the supply voltage to gauge the remaining margin and provide information to the degradation model. Each calibration routine runs predesigned tests targeted to stress many critical paths [Wagner and Bertacco 2008]. The test is repeated for several voltages, all lower than nominal, at the nominal frequency to estimate the margin. One calibration routine may require several seconds of execution, but the impact is insignificant since this calibration occurs only once every several days.

A hybrid approach that exploits advantages of both the hardware and software approaches can also be used [Basoglu et al. 2010]. In the hybrid approach, hardware sensors provide more frequent feedback for NBTI degradation measurement. Additionally, extensive software calibration, which is performed infrequently, improves the accuracy of the measurement. The hardware sensors improve the energy-savings 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 limitations of the hardware sensing approach regarding critical paths. In the hybrid approach, software calibration can be very infrequent, and thus more extensive. More extensive calibration allows the system to better identify critical paths and further reduce the degree of conservatism.

2.4. Positive Bias Temperature Instability (PBTI)

BTI includes NBTI in p-MOSFETs and PBTI in n-MOSFETs. PBTI degrades the lifetime of n-MOSFETs by increasing the magnitude of V_{th} when gates are positively biased. It is well explained by the reaction-diffusion (R-D) model [Sa et al. 2002] as NBTI is. In sub-45-nm process technologies, high-k metal-gate transistors are widely used due to low leakage and high performance [Hicks et al. 2008]. A drawback, however, is that PBTI becomes significant in high-k meal-gate transistors, which was negligible when SiO_2 was used as dielectric and gate. The existing studies on PBTI degradation specifically target the designs in high-k metal-gate process. Thus, most studies are circuit-level techniques and related to SRAM design. The techniques reduce the impact by both NBTI and PBTI at the same time, rather than deal with only PBTI.

NBTI and PBTI for various field effect transistors (FETs) were compared in Zafar et al. [2006]. This study showed that NBTI degradation in SiO_2/NiSi and SiO_2/HfO_2/NiSi FETs is same as that in conventional transistors. Meanwhile, PBTI impact considerably increased, becoming a more severe reliability issue than NBTI. BTI characterization on 45-nm, high-k metal-gate transistors is presented and degradation mechanism is discussed in Pae et al. [2008], who optimized the processing condition to achieve BTI degradation that is comparable to or better than with SiO_2 dielectrics. On the optimized process, NBTI degradation was interface driven as it is in SiON dielectrics, while PBTI degradation is attributed to electron trapping in the high-k bulk and its interface layer with SiON. Zhao et al. [2011] also addressed that the PBTI degradation in high-k metal-gate transistors is primarily affected by trapping and detrapping events. A lifetime prediction based on a single-defect model was presented with experimental data that supported the validity of the model.
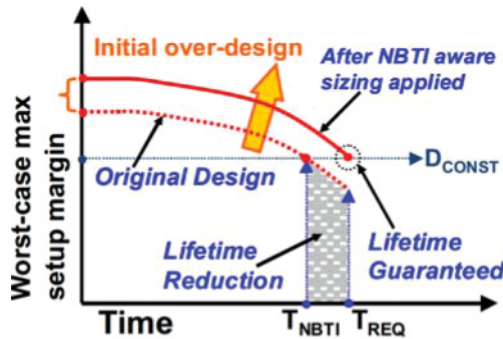
Fig. 4. Lifetime guaranteed by reliability-aware gate sizing [Kang et al. 2006].

In sub-45 nm technologies, SRAM cells are severely affected by BTI since SRAM static noise margin is highly sensitive to device mismatch that BTI causes [Hicks et al. 2008]. Krishnappa and Mahmoodi [2011] addressed that eight-transistor (8T) or ten-transistor (10T) SRAM cells should be used rather than traditional six-transistor (6T) SRAM cells in future high-k metal-gate technologies. The supply voltage in 8T and 10T SRAM cell design can be reduced by less than half of that in 6T SRAM cell design. Thus, both NBTI and PBTI degradation is considerably mitigated. Another SRAM cell design to mitigate both NBTI and PBTI degradation at the same time was proposed in Li et al. [2011] for high-k metal-gate process. The SRAM cell has four internal gates, all of which are recovered in a proactive recovery mode, thereby slowing down the V_{th} shift.

Power-gated cells are widely used in most state-of-the-art SRAMs to reduce leakage current in standby and sleep mode. An optimal source biasing of power-gated SRAM in standby mode is proposed in Pushkarna and Mahmoodi [2010]. The significant decrease in standby bias voltage mitigates both NBTI and PBTI degradation. A comprehensive analysis on the BTI impact on power-gated SRAM array is presented in Yang et al. [2011], who also cover the impact on SRAM cell stability, margin, and performance by BTI and propose BTI-tolerant sense amplifier structures.

3. CIRCUIT-LEVEL TECHNIQUES

Lifetime reliability has been traditionally treated as a manufacturing problem. Many studies have been conducted to enhance the lifetime of microprocessors at the circuit-level. The existing techniques can be classified into two groups: design-time techniques and adaptive techniques.

3.1. Design-Time Techniques

As process technology shrinks into the nanometer regime, designers have to add pessimistic timing margins to the circuit to stave off timing violations due to transistor aging. This is the basic design technique for lifetime improvement, which is called guard-banding or overdesign. This wastes performance at the beginning of a part's lifetime.

3.1.1. Gate/TR Sizing. One of the early approaches to enhance lifetime reliability is optimal gate sizing. A gate-sizing method that uses a modified Lagrangian Relaxation (LR) algorithm was proposed in Paul et al. [2006]. It computes the optimal gate size while considering the impact of NBTI degradation. Figure 4 explains the main idea of the proposed method. The dotted line represents the setup timing margin of the

original design. It decreases with time, and eventually the setup time falls below the specified timing constraint (D_{CONST}) after a certain stress period (T_{NBTI}), thereby failing the design earlier than intended. The solid line in Figure 4 represents the setup time margin in which transistors are sized up so that the circuit can ensure functionality even when the demanded lifetime T_{REQ} is expended. The results from Paul et al. [2006] showed that an area overhead is 8.7% on average for ISCAS benchmark circuits implemented with 70nm technology to ensure a lifetime of three years.

Another transistor-level sizing technique was proposed in Kang et al. [2006], who applied a more sophisticated method using separate sizing factors for p-MOSFETs and n-MOSFETs rather than sizing both at the same time. This reduces unnecessary slack in the n-MOSFET network that is not influenced by NBTI. As a result, the overall area overhead is lowered compare to the work by Paul et al. [2006]. Experiments showed that the average overhead reduction was about 40% compared to the method of Paul et al. [2006].

Based on the observation that the probability of p-MOSFET transistors being stressed is nonuniform across all transistors in a circuit, a mathematical formulation of NBTI-aware gating sizing was proposed in Yang and Saluja [2007]. The delay margin of each transistor is assigned with respect to its expected probability of being stressed. The gate sizing in Kang et al. [2006] is processed individually for each part of the circuit and if a timing violation is found, the size of the corresponding transistor is increased. This may lead to oversizing. Meanwhile, the technique in Yang and Saluja [2007] handles the timing problem as a single optimization for a whole circuit, enabling more accurate and area-efficient gate sizing.

Recently, a gate sizing that considers both NBTI and oxide breakdown (OBD) was proposed [Roy and Pan 2014]. The developed static timing analysis (STA) engine and gate sizer embed a piecewise linear model of rise delay/rise slew with NBTI degradation. This work is the first to consider that the slew degradation due to NBTI could increase the fall delay of the inverting gates in the next stage.

3.1.2. NBTI-Aware Synthesis. Another method is to address NBTI degradation problems during the technology mapping of the logic synthesis [Kumar et al. 2007]. The basis of this approach is that different standard cells show different NBTI sensitivities regarding their input signal probabilities. Using this observation, Kumar et al. [2007] revise the standard cell library by considering extra signal probability dependency. The revised library is used during logic synthesis to decrease the effect of NBTI. The revision of the library may cause area overhead. Kumar et al. [2007] showed that there was an average of a 10% less area overhead than that of the worst-case logic synthesis. Wang et al. [2007b, 2009] examined the use of input vector control (IVC) to decelerate aging and found that NBTI degradation can be reduced using IVC by 30% on average. The actual implementation of IVC can be done by either adding MUXes to the inputs or including scan chains. The drawback of IVC is that the input vector can control only a very small portion of the gates in the circuit in most cases. Wang et al. [2007b, 2009], nevertheless, predicted that small gate size and high temperature can enhance the merit of IVC in future technologies.

Internal Node Control (INC) is more powerful in terms of controllability than IVC, allowing much reduction in delay induced by NBTI. In Bild et al. [2009], the inputs to each gate can be directly controlled so that static NBTI degradation can be reduced. A mixed integer linear program was introduced as an optimal solution. Since the problem is NP-complete, a linear-time heuristic was also proposed to reduce time and complexity to obtain a solution. The result of the optimal placement of INC provided that 26.7% reduction is achieved for the ISCAS85 benchmarks in delay induced by NBTI over a lifetime of 10 years.

An aging-aware logic synthesis approach is proposed Ebrahimi et al. [2013]. As the name of the technique implies, it takes not only NBTI but also PBTI and HCI into account to build the aging model. The design timing is optimized with respect to postaging delay in a way such that all paths reach the assigned guard band at the same time. In this regard, in an iterative process, after computing the postaging delays, the lifetime is improved by putting tighter timing constraints on paths with higher aging rates and looser constraints on paths that have less postaging delay than the desired guard band. Ebrahimi et al. [2013] showed that the proposed approach improves circuit lifetime by more than three times on average with negligible area overhead by implementing the method on top of a commercial synthesis tool chain.

In Lai et al. [2014], a clock-gating methodology, BTI-Gater, is proposed to reduce NBTI- and PBTI-induced clock skew and imbalanced degradation, which conventional clock gating may cause. The R-D model for NBTI and PBTI degradation was calibrated using a commercial process technology. It showed that the PBTI impact on high-k metal-gate transistors is significant. The integrated clock-gating, cell-based implementation reduced the BTI-induced clock skew and leakage power.

3.2. Adaptive Techniques

Since NBTI degrades the circuits for a very long time, the effectiveness of the design-time techniques introduced in Section 3.1 can be easily undermined. Adaptive techniques, however, monitor the degradation over time and are capable of coping with each lifetime phase differently. SRAM cells suffer from heavy NBTI stress due to unbalanced duty cycle ratio. Jin and Wang [2013] investigated the lifetime behavior of cache lines and proposed a duty cycle balancing scheme for the instruction cache. The effectiveness of design-time techniques was compared with that of adaptive techniques, such as FBB and AVS, in Kang et al. [2008]. The adaptive techniques efficiently reduced the impact of NBTI even when combined with the impact of process variations.

3.2.1. Forward Body Biasing (FBB). Body biasing is used to control the threshold voltage V_{th} of a MOSFET by altering the substrate body voltage. It has been utilized mainly to decrease leakage power or to hide performance asymmetry due to process variations. Body biasing can be either forward or backward. FBB is used to decrease the threshold voltage, thereby speeding up the circuits. Decreasing the threshold voltage may cause exponential increase of the subthreshold leakage [Narendra et al. 2003]. Reverse body biasing (RBB), on the other hand, can reduce subthreshold leakage by increasing the threshold voltage, costing performance loss. The V_{th} of a short-channel MOSFET in the BSIM model [Ko et al. 1993; Liu et al. 1993] is described as:

$$V_{th} = V_{tho} + \gamma(\sqrt{\varphi - V_{bs}} - \sqrt{\varphi}) - \eta V_{DD} + \Delta V_{NW}, \tag{6}$$

where

$$\gamma = \frac{\sqrt{2\varepsilon_s q N_A}}{C_{ox}}. \tag{7}$$

V_{tho} is the threshold voltage when $V_{bs} = 0$. V_{bs} represents the difference between the body and the source voltages. φ represents the surface potential and η represents the barrier lowering coefficient because of drain, V_{DD} is the supply voltage, and ΔV_{NW} is a constant representing the effect of narrow width [Lee and Kim 2011]. Although it is not the original purpose of introducing FBB, it can be used to mitigate NBTI degradation since it adaptively lowers the threshold voltage during the lifetime of a circuit. There have been several attempts to alleviate the impact of NBTI using FBB.

A reliability monitor was proposed in Qi and Stan [2008], which trails the NBTI effect and alleviates NBTI degradation by forward biasing. NBTI monitoring uses a

ring oscillator–based structure like other techniques introduced in Section 2.3. A design for a reliability scheme was proposed, which includes a self-adjustable threshold voltage (SATV) [Khan et al. 2011]. The SATV basically lowers the threshold voltage by body biasing, but applies no body biasing in the absence of NBTI.

Reliability has become a major design challenge to SRAM designers. Mostafa et al. [2011] proposed an adaptive body bias (ABB) circuit with low area overhead to make up for the impact of NBTI and process variations to enhance the SRAM reliability. The ABB circuit is composed of a sensor and a controller. Mostafa et al. [2011] conducted postlayout simulations using STMicroelectronics 65nm CMOS technology. The results show that the read failure probability is reduced to 0.05%, and the static-noise margin degradation is reduced to 2.6% for 10 years of lifetime. The soft error immunity of the SRAM cell was also enhanced by the reduction of the critical charge degradation.

The aggressive scaling toward the deep submicron technology has grown the parameter variations, such as channel length and threshold voltage. These parameter variations are regarded as one of the most critical issues of design for future technologies [Bowman et al. 2002; Borkar et al. 2003; Masuda et al. 2005]. There are two categories of process variation: within-die (WID) and die-to-die (D2D) variations. Considering WID variations, devices on the same die can have different parameters. However, it is assumed that in D2D, devices on the identical die show the identical parameters, while devices on different dies have different parameters [Masuda et al. 2005]. Mostafa et al. [2012] expanded and evaluated their body biasing technique for a CMOS technology transistor model. An optimized microarchitecture design that reduces design guardband by combining NBTI recovery mechanism and process variation mitigation was proposed by Fu et al. [2008]. The optimization is based on the positive interplay between process variation and NBTI.

3.2.2. Adaptive Voltage Scaling. Voltage scaling is used to control the supply voltage, V_{DD}, in order to raise the operation speed of a circuit or to lower the power consumption. The propagation delay of transistors can be approximated by Equation (4). The increase in supply voltage causes the decrement of the propagation delay, thus using AVS can recover the increased delay due to NBTI. However, there are drawbacks using AVS in order to alleviate the effect of NBTI. Dynamic power is proportional to the square of the supply voltage, thus increasing the supply voltage results in growth in power consumption. Additionally, the increase in power results in the increase of circuit temperature, that is, temperature is highly related to power consumption, which can be represented as [Cengel 1997]:

$$T = T_a + PR_{th},\qquad(8)$$

where T represents the circuit temperature, T_a is the ambient temperature, P is the power consumption, and R_{th} is the thermal resistance of the chip. Voltage scaling to alleviate NBTI degradation should be used with caution due to the close correlation between temperature and power consumption. The performance degradation can be hidden by increasing V_{DD}, but that increases power consumption and temperature, which may accelerate NBTI degradation.

A scheduled voltage scaling that minimizes and compensates the NBTI degradation is proposed in Zhang and Dick [2009]. General guard-banding fixes operating voltages at the early design stage, while, the proposed scheduled voltage scaling increases the operating voltage gradually during operation. This allows V_{DD} to be always kept lower than necessary by guard banding. A decrease in operating voltage or temperature alleviates the wear rate due to NBTI [Vattikonda et al. 2006]. The wear rate is reduced directly by cutting down the voltage, and indirectly by the correspondingly reduced temperature and power consumption. The proposed voltage scaling takes several
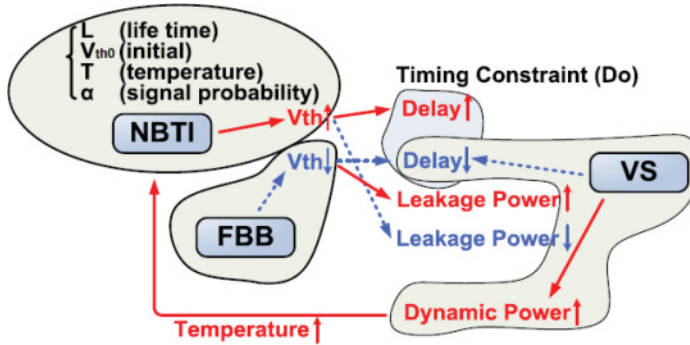
Fig. 5. Graph representing how NBTI, FBB, and VS influence the design parameters [Lee and Kim 2011].

factors into account to generate an ideal voltage schedule: the accumulation of NBTI degradation, effect of leakage power consumption, and temperature. Vattikonda et al. [2006] showed that by using gradually increasing voltage scheduling, the lifetime of a 45-nm-process chip is enhanced by 46% compared to that of the conventional method, without any growth in power consumption or temperature.

Chen et al. [2012] proposed a variation-aware supply voltage assignment (SVA) technique that merges dual supply voltage assignment and dynamic scaling. First, gates are divided into two categories: a high supply voltage gate set (HVGS) and a low supply voltage gate set (LVGS). NBTI-aware critical gate is defined as a gate such that the slack of the gate is smaller than a given threshold value. HVGS includes all NBTI-aware critical gates and their predecessors, while LVGS includes all the remaining gates. Afterwards, high supply voltage and low supply voltage are calculated to minimize power consumption under the NBTI-aware timing constraint.

Varying the supply voltage of a circuit using AVS also causes BTI degradation to vary over the circuit's lifetime. This presents a new challenge for margin reduction in conventional signoff methodology, which characterizes timing libraries based on transistor models with precalculated BTI degradations for a given IC lifetime. A BTI-aware signoff that accounts for the use of AVS during IC lifetime was first proposed in Chan et al. [2013]. Based on simulations and analysis of performance degradation due to BTI in the presence of AVS, a rule of thumb for chip designers to characterize an aging-derated standard cell timing library that accounts for the impact of AVS was proposed to avoid both overestimation and underestimation.

3.2.3. Hybrid Techniques. FBB manipulates the threshold voltage, while AVS controls the supply voltage, but both FBB and AVS can mitigate NBTI degradation. The two techniques are somehow dependent on each other, and NBTI degradation can be alleviated when both are utilized at the same time. Lee and Kim [2011] proposed a fine-grained technique to minimize power consumption and reduce NBTI degradation by simultaneously utilizing FBB and AVS. They conceptually showed the impacts on the threshold voltage, delay, leakage power, dynamic power, and temperature by FBB, voltage scaling (VS), and NBTI in Figure 5. The red solid arrows represent the flows that result in the increase of the parameters. Following the red arrow started from NBTI, the threshold voltage of the device increases due to NBTI and eventually increases the delay of the device. The blue dotted arrows represent the flows that result in their decrease. Following the blue dotted arrow starting from FBB, it lowers the threshold voltage; as a result, the delay is reduced. VS also reduces the delay, but this results in the increase in dynamic power consumption. Since the impact by FBB and VS on NBTI cannot be clearly seen, a technique that combined FBB and AVS to keep

the optimum performance of an aging circuitry is also proposed in Kumar et al. [2011], who show that the combination of FBB and AVS outperforms NBTI-aware synthesis. In addition, they present a hybrid approach that exploits the advantages of both FBB and NBTI-aware synthesis.

4. ARCHITECTURAL TECHNIQUES

Aging due to NBTI is accelerated at high temperatures, and temperature depends on the power consumption of devices. Therefore, low-power design techniques can efficiently manage the lifetimes of microprocessors. Also, exploiting spare components in architecture level can extend lifetime reliability.

4.1. Power Gating

A power gating is one of the popular low-power design techniques that was originally used to lessen power consumption by cutting off power supply to idle functional blocks [Tsai et al. 2004]. The implementation of power gating typically includes a sleep transistor between the functional block and the power supply. Static and dynamic power consumption can be reduced by turning off the sleep transistor. Since it decreases the power consumption, temperature drops, which leads to the alleviation of NBTI degradation. In addition, stress time is reduced while recovery time is increased. The weakness in employing power gating is its time consumption in the process of power up or down. Therefore, the power gated functional block is inactive for some time in the process of power up or down, which can lead to damages in performance. However, in multicore processors not every core is always activated, thus power gating can be effective by switching off unused cores. For this reason, power gating is often used jointly with other system-level solutions [Karpuzcu et al. 2009; Basoglu et al. 2010].

Calimera et al. [2009] showed how power-gating provides a natural method of reducing NBTI degradation. The transistor delay changes due to the increase in threshold voltage due to NBTI as

$$\text{Delay}'(t) = \text{Delay}\left(1 + \frac{\Delta V_{th}}{V_{GT} - \Delta V_{th}}\right), \tag{9}$$

where

$$\Delta V_{th} = K(\beta t)^{1/4}, \tag{10}$$

where $V_{GT} = V_{gs} - V_{th,0}$ and $V_{th,0}$ is the nominal threshold voltage at time 0. If a system is power gated, the delay becomes

$$\text{Delay}_{\text{gated}}'(t) = \text{Delay}_{gated}\left(1 + \frac{K(\beta(1 - P_{sleep})t)^{1/4}}{V_{GT} - K(\beta(1 - P_{sleep})t)^{1/4}}\right), \tag{11}$$

where $\text{Delay}_{\text{gated}} = \text{Delay}\,(1 + \gamma)$ is the delay of the power-gated circuit at time 0 and P_{sleep} is the probability of the sleep signal. Higher values of γ imply smaller sleep transistors. From a first-order analysis, Calimera et al. [2009] showed that delay is reduced using power gating.

A compliant numerical model for NBTI degradation is proposed in Chan et al. [2011], which is argued to estimate more efficiently the effect of architectural techniques on NBTI degradation. The model contributes a numerical solution to the reaction–diffusion equations depicting NBTI degradation that have been demarcated to design the effect of several architectural techniques: DVFS, averaging effects across logic paths, activity management, and power gating. The effects of some architectural techniques were compared by Chan et al. [2011], including power gating, using the proposed numerical model.
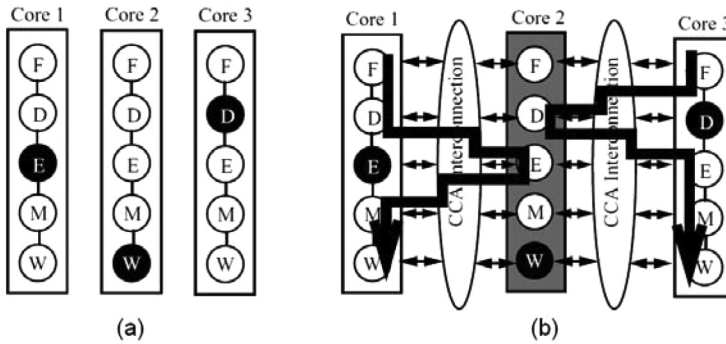
Fig. 6.   (a) General multicore processor architectures; (b) core cannibalization architecture [Romanescu and Sorin 2008].

4.2. Self-Repair

One traditional approach to improve lifetime is to self-repair by utilizing spare components. Although spare components spend hardware resources without performance benefit during fault-free operation, they can enhance lifetime reliability. A physical copy for lifetime reliability improvement is introduced in Srinivasan et al. [2005b]. There are two approaches that leverage structural redundancy: first, in the physical copy, extra architectural structures that were assigned as spares were included to the processor. Extra structures are used only when the given structure has failures. The second technique is to use the existing architectural extras for reliability improvement. The failed structures are not utilized while the whole system maintains its functionality all the same, at a lower performance. The recent shift of architecture from multicore to many-core significantly increases the number of integrated cores. For many-core processors, introducing core-level redundancy is an efficient method to reduce the aging problem. There can be a number of approaches that exploit redundant cores, and they have different implications on the NBTI degradation. This second techniques will be presented in Section 5.3.

The granularity of the redundancy can be finer than a whole core. A multicore architecture that improves lifetime reliability by partial sharing is proposed in Romanescu and Sorin [2008]. If there is just one hard fault in a block, it makes the core useless although the remaining blocks of the core are fault-free. The core-level self-repair schemes are basically to use only the remaining cores and not to use the core that has one or more hard faults. This core shutdown (CS) solution does not fully utilize fault-free circuitry. Romanescu and Sorin [2008] proposed the Core Cannibalization Architecture (CCA), an efficient self-repair scheme for multicore processors that consist of simple cores. The main idea is that cores can be divided into several blocks according to pipeline stages and the blocks can be the unit of redundancy. In Figure 6, black circles represent faulty blocks. Although one pipeline stage for each core is faulty, all three cores are faulty, resulting in failing the whole system in a general multicore architecture. In the CCA, Core 1 and Core 3 partially share Core 2 so that two out of three cores can work. Despite the performance loss, the lifetime of the multicore processor is lengthened. To support the finer redundancy, there may be a tiny performance overhead. However, Romanescu and Sorin [2008] insist that the lifetime reliability enhancement is much more important than the overhead.

5. SYSTEM-LEVEL TECHNIQUES

Circuit-level techniques to enhance microprocessor lifetime are based on overdesign, which guard bands designs with estimation of worst-case temperature and
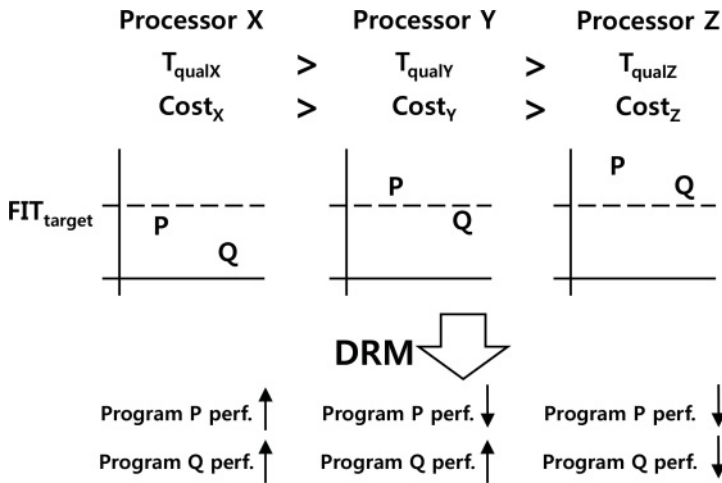
Fig. 7. Motivation for DRM where the dashed line representing target FIT, the required failure ratio [Srinivasan et al. 2004].

processor utilization. However, circuit-level techniques often show high reliability and long lifetimes since applications generally operate under a lower temperature and lower utilization than the worst case. Circuit-level techniques, which are not application-aware, tend to be overly conservative, resulting in unnecessary increases in cost and performance loss. Dynamic reliability-aware solutions in the system level can compensate for these defects. Aging is highly dependent on utilization and the operating temperature of a device. System-level techniques exploit the application runtime behavior to improve lifetime reliability, thus cost and performance can be improved.

Figure 7 illustrates the motivation of the dynamic reliability management (DRM) proposed in Srinivasan et al. [2004, 2005a]. FIT stands for failures in time; the value smaller than FITtarget means that the reliability requirement is met. There are three processors that show a trade-off relationship between cost and lifetime reliability. Processor X is the most expensive to ensure the required lifetime reliability, and Processor Z is the cheapest. Assume that there are two application programs, P and Q. The programs will have dissimilar FIT on the three processors since the reliability constraints of the processors are unequal. In Processor Z, both programs P and Q do not attain the FITtarget value. In Y, Q attains the FITtarget value, but P does not. In other words, Processors Y and Z are underdesigned for the two programs. Meanwhile, Processor X is overdesigned. Thus all application programs driven by Processor X comply with the FITtarget value, but their FITs are excessive.

There are two points at which DRM can be utilized. The first case is the reliability enhancement of underdesigned processors, for instance, the cases that Y operates program P and Z operates either P or Q. Though Y and Z are less expensive than X to meet the lifetime reliability, they can fail prematurely and cannot meet the reliability requirements. Using DRM, the reliability of Processors Y and Z can be enhanced in order to alleviate temperature, voltage, and frequency during operation. The second usage of DRM is the performance improvement of overdesigned processors. The example shown in Figure 7 is the case that programs P and Q are on Processor 1 and program Q is on Processor 2. DRM can be used to employ the reliability margin and derive an additional performance upgrade if a cooling solution can support it.
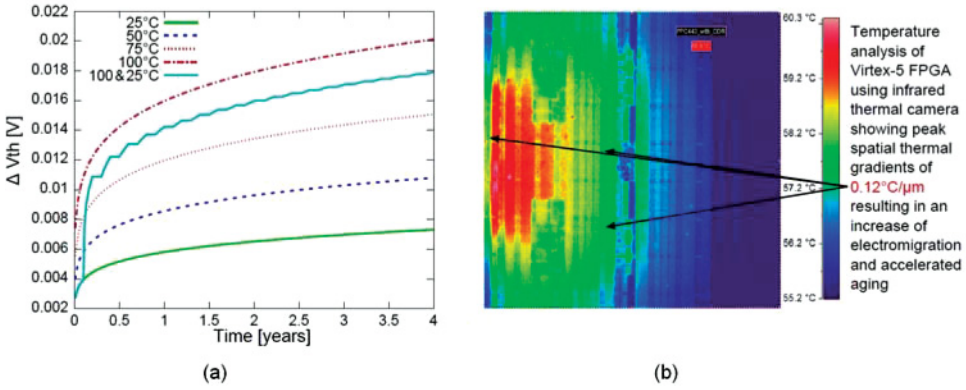
Fig. 8.   Impact of temperature on aging: (a) NBTI-induced threshold voltage shift at various temperatures;
(b) a thermal image of an FPGA chip [Henkel et al. 2013].

Aging is highly temperature-dependent. The dependency is expressed through Ar-
rhenius's Law [White 2008]. An aging effect λ_{EFF} has the property

$$\lambda_{EFF} \propto e^{\frac{-E_a}{kT}}, \tag{12}$$

where T represents the temperature and k represents Boltzmann's constant, respec-
tively. The activation energy E_a is specific for a certain aging process. Figure 8 illus-
trates the impact of temperature on the threshold voltage shift. It clearly shows that the
increase in temperature accelerates the threshold voltage shift. Since DRM can take
the current temperature of a device into account, it can improve lifetime reliability
more efficiently than circuit-level static techniques.

5.1. Scheduling/Load Balancing

Utilization causes wear out, eventually affecting lifetime reliability. Equalizing the
utilization of functional units in a microprocessor can improve its lifetime reliability.
One of the early-stage works in NBTI-aware scheduling was proposed in Siddiqua and
Gurumurthi [2009], who evaluated the impact of various instruction schedulings on
NBTI degradation and proposed NBTI-aware instruction scheduling. Another instruc-
tion scheduling is proposed in Siddiqua and Gurumurthi [2010], who combined instruc-
tion scheduling with power gating, which is one of the circuit-level techniques. This
multilevel approach was shown to more effectively reduce the design guard band than
instruction scheduling. Oboril and Tahoori [2012b] propose an aging analysis frame-
work that includes clock gating, power gating and NBTI-aware instruction scheduling.
Oboril et al. [2013] propose a cross-layer approach. Instructions are identified as either
critical or noncritical based on the worst-case circuit-level delay and how often they
occur in application level. It was shown that cross-layer instruction scheduling extends
transistor lifetime compared to a load balancing policy.

Microprocessor architecture has shifted to multicore processors, even to many-core
processors. In homogeneous multicore processors, task allocation and scheduling could
make huge differences in terms of performance, energy efficiency, and other design
aspects. Aging is one of these aspects. If a core handles a larger number of tasks than
other cores, the core will age faster than the others, and one faulty core can make
the whole multicore processor obsolete. Therefore, task allocation and scheduling must
take into account the impact of NBTI.

Energy-efficient task allocation and scheduling under lifetime reliability constraints
is proposed in Huang and Xu [2010a]. The idea starts from the fact that embedded

systems usually have various execution modes. Huang and Xu [2010a] identified a set of optimal task allocation and scheduling for each operation mode regarding lifetime reliability and energy consumption. This is called a single-mode solution. The method is proposed to derive the best mixture of these single-mode solutions to diminish the energy expenditure of the system while complying with a given lifetime reliability restriction. Huang et al. [2011] propose an analytical model for lifetime reliability of multicore processor platforms when performing tasks periodically. Based on the model, they propose a new task allocation and scheduling algorithm that can reduce aging due to NBTI, using simulated annealing. In order to speed up annealing, several techniques are introduced by reducing design space exploration.

Devices wear out under stress, but they recover to a certain level once the negative bias is eliminated. By exploiting the recovery, aging can be slowed further. A new design framework for multicore processors that takes device wear-out impact into account is proposed in Sun et al. [2009], who propose a new NBTI-aware system workload model built on the device fractional NBTI model. A dynamic tile partition algorithm was developed to maintain the workload among busy cores, providing the opportunity for stressed cores to relax. Experimental results on 64 cores showed that the proposed dynamic tile partition improves the yield of multicore systems. The core failure number was increased by 20%, and MTTF was extended by 30% with less than 6% performance degradation.

The usage of large-chip multiprocessor (CMP) systems has become common. Although technology scaling aggravates lifetime reliability and issues such as process variation become more severe, the heterogeneity of large CMPs can be a potential opportunity. Maestro, described in Feng et al. [2010], is the CMP system with enhanced lifetime reliability based on recognition. By taking advantage of sensor feedback, Maestro formulates a wear out–centric scheduling that accomplished both global and local wear leveling. The proposed scheduling prevented early core failures, enhancing the expected CMP lifetimes up to 38%, and enhancing the throughput of a 16-core CMP up to 180%.

5.2. Dynamic Voltage and Frequency Scaling

Dynamic voltage and frequency scaling (DVFS) and power gating are the most popular low-power design techniques. Like power gating, DVFS also can efficiently manage the lifetimes of microprocessors.

A technique named Facelift is proposed by Tiwari and Torrellas [2008], which conceals aging due to the NBTI effect through temperature-aware scheduling to each core of a multicore processor. The basic idea of Facelift is to allocate high-temperature jobs to the fast cores and low-temperature jobs to the slow cores, respectively. This allocation minimizes aging of the chip by keeping the slow cores cooler. Also, Facelift makes chip-wide alterations to V_{DD} or V_{th} at critical times to compensate for the effect of the alterations on critical path delays and on aging rate.

DVFS lowers the supply voltage and/or frequency using given timing slacks; as a result, power consumption decreases. An NBTI-aware DVFS framework is proposed by Basoglu et al. [2010], which simultaneously reduces energy consumption and increases lifetime reliability. The proposed framework utilizes real-time degradation data collected from temperature sensors. A core-level DVFS and OS-controlled workload mapping based on core status were also proposed. The OS is informed of the degradation status of each core, and can map threads so that sturdy cores with low-threshold voltages work more than the weaker cores with high-threshold voltages. This gradually equalizes core lifetimes, and extends overall processor life. Figure 9 illustrates the baseline of the proposed multicore framework. The colored modules make difference from the general DVFS framework. The NBTI calibration and voltage
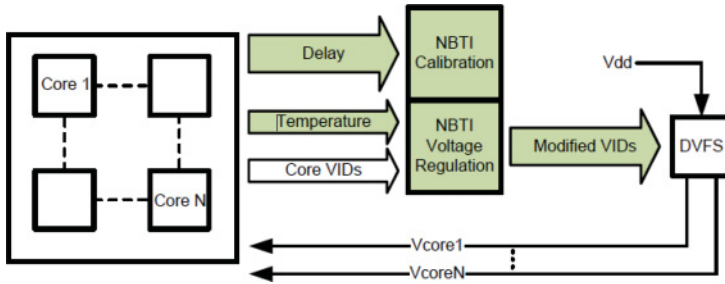
Fig. 9.   Baseline multicore processor for NBTI-aware DVFS [Basoglu et al. 2010].

regulation modules are informed of each core's degradation status and use this information when determining each core's supply voltage and frequency.

Karakonstantis et al. [2010] introduce a self-consistent estimation model of NBTI degradation considering the overall impact of interdependent parameters, such as supply voltage and temperature, on the lifetime reliability. Using this model, they showed that a circuit using a lower supply voltage is able to show better lifetime reliability than one using a higher supply voltage. Scaling down the supply voltage, however, may cause failures due to the increase in delay, thus voltage scaling has to be carefully determined. The impact on NBTI degradation of supply voltage was decided in advance and kept in a look-up table. The proposed method amends potential errors by using a slow clock, and alleviates these by dynamically using a higher supply voltage.

A general framework that jointly optimizes multiple self-tuning parameters over lifetime of a system is proposed by Mintarno et al. [2011]. The framework dynamically adjusts the self-tuning parameters according to time-varying performance demands and estimated system aging: It jointly optimizes trade-offs between the affirmative influence of lowering the temperature on circuit aging, leakage power, and delay. The negative influence of power consumption for lowering temperature is also optimized.

A dynamic runtime adaptation approach is proposed in Oboril and Tahoori [2012a]. It is a fine-grained DVFS as opposed to prior DVFS techniques for mitigating NBTI degradation. A coarse-grained DVFS tends to be activated after aging has been accumulated to a certain level. Moreover, its static nature prevents it from instantly responding to dynamic events with mutational environmental states, performance, or power demands. Therefore, a fine-grained DVFS is more desirable to prolong lifetime reliability, to decrease power consumption, and to lower temperature, while sustaining the demanded performance. The proposed fine-grained DVFS supports several time adjustments per second, which is reasonable in real devices. The expert system decides the arrangement using both voltage and frequency for the time duration after each time frame by utilizing the information on both previous and present system status, the speculated system performance, and user-specific constraints.

5.3. Self-Repair

By using existing architectural extras, lifetime reliability can be improved. For multi- or many-core processors, introducing core-level redundancy is an efficient method for reducing the aging problem. To maximize the effectiveness of using redundant cores in many-core processors, it is critical to distinguish the lifetime reliability with various utilizations. Huang and Xu [2010b] propose an analytical scheme that considers the amount of work and associated temperature variations in order to overcome the difficulty. The proposed scheme is used to analyze the reliabilities of many-core processors with different redundancy arrangements. The first arrangement is the Gracefully

Degrading System, which initially regards all n cores as busy cores. When there is a faulty core, the system is rearranged so that the (n−1) good cores can handle the workload. This procedure stops when there are only m good cores. When the left good cores are less than m, finally, the whole system is regarded as faulty. Good cores alternatively operate in busy mode and idle mode to equalize their aging status. This arrangement is named the Processor Rotation System (PRS). Huang and Xu [2010b] introduced another configuration called the Standby Redundancy System (SRS), in which initially (n−m) cores are in idle mode and m cores are in busy mode, respectively. If a faulty component is detected, the system substitutes the faulty one with a spare one and uses the spare code. The difference is that the goal of PRS is to stabilize the aging of all cores, but the SRS does not care about the balance of cores, only about the whole system's lifetime.

A performance acceleration of many-core processors under power constraints is proposed in Karpuzcu et al. [2009]. Although the main object of this work is not to improve the lifetime of processors, it includes the use of core-level redundancy. In the proposed architecture, not all cores are activated at the beginning of the device lifetime. Some cores, called throughput cores, are dedicated to running the parallel parts of the given applications. Other cores, called expendable cores, are devoted to operating the sequential parts. Expendable cores run sequential parts at elevated supply voltages for a significantly shorter lifetime compared to throughput cores. If an expendable core fails, it is discarded and substituted with a neighboring expendable core. This substitution changes the boundary of currently activating cores, which led to the idea of naming the method BubbleWrap.

By exploiting spare cores, the wear out of a single core in many-core CMPs may not be catastrophic for the entire system. A single fault in Network-on-Chip (NoC) fabric, however, could render the entire chip useless, as it can result in serious damage, such as protocol-level deadlocks, or partition vital components, such as the memory controller, away. The critical path for HCI- and NBTI-induced wear out for actual stresses caused by real workloads is modeled by Kim et al. [2013]. The models are applied on the interconnect microarchitecture. Kim et al. [2013] observed that wear out in the CMP on-chip interconnect is correlated with a lack of load observed in the NoC routers rather than with a high load. Based on the observation, a novel wear out–decelerating scheme was also proposed in which routers under low load have their wear out–sensitive components exercised, without significantly impacting cycle time, pipeline depth, area, or power consumption.

As multicore or many-core processors are widely adopted, more studies have been conducted on core-level redundancy rather than finer-grained redundancy. Systems should be reconfigurable in order to support functional unit-level redundancy; one functional unit has to be physically connected to other functional units in the same core but also to some functional units in other cores. This leads to longer wires and more complicated logics. On the other hand, core-level redundancy can be more flexibly and effectively implemented. No extra wires are required just to support core-level redundancy, but using power gating and proper task scheduling and assignment can implement core-level redundancy.

Three-dimensional (3D) integration technology has been accepted as one of the solutions to the problems faced by traditional two-dimensional (2D) integration technology. Although 3D die stacking offers many advantages, some existing problems may be exacerbated. One of the challenges is the thermal crisis. Temperature is a major factor in NBTI degradation, thus more intensive studies are required to mitigate NBTI impact in 3D ICs. However, few studies have been performed. Recently, a lifetime enhancement of multicore processors through 3D resource sharing was proposed in Strikos [2013]. In the proposed 3D multicore processors, cores are vertically stacked and allow

Table I. Comparison of NBTI Mitigation Techniques across Design Level

Design level	Technique	Major control factor	Control detail
Circuit level	Gate/TR Sizing	Slack	Sizing gate/TR based on timing analysis
	NBTI-Aware Synthesis	Signal Probability	Input Vector Control (IVC) or Internal Node Control (INC)
	FBB	Delay	Lower V_{th}
	AVS		Lower V_{dd}
Architectural level	Power Gating	Time under stress	Reducing time device under stress
	Self-Repair	-	Hardware redundancy
System level	Scheduling /Load Balancing	Time under stress	Reducing time device under stress
	DVFS	Delay	Lower V_{dd}
	Self-Repair	-	Exploiting redundancies in multicore processors

their resources to be shared. Since the physical distance between cores is shorter in 3D systems due to wafer thinning, the sharing can be more effective than that in 2D systems. The proposed approach is based on resource pooling, which enhances the lifetime reliability of the multicore system by dynamically adapting the core resources without redundancy or performance loss. Core resuscitation is a fine-grained reconfiguration around faulty components that prevents decommissioning an entire core because a single resource is faulty. Resource salvaging is a technique in which functioning units of decommissioned cores can be assigned to the surviving cores to improve both their performance and lifetime. Experimental results show that resource pooling results in the most significant improvement in performance and lifetime reliability simultaneously.

6. CONCLUSIONS

In this article, we introduced various techniques to enhance the lifetime reliability of microprocessors by mitigating NBTI degradation. The existence of NBTI has been known for decades, but its impact has been ignored compared to other factors, for example, HCI. Processors undergo repeated stress and recovery phases across their lifetime, and the threshold voltage and carrier mobility gradually shifts. As a result, delay is increased so that the processor cannot meet performance requirements. Mitigation techniques can be effective if they are based on an exact NBTI model. Beyond the traditional delay modeling, recent NBTI models include other design parameters such as process variations, temperature variations, and HCI.

Mitigation techniques can be classified according to design level. Table I compares the NBTI mitigation techniques across design levels. Circuit-level techniques include design-time techniques such as transistor sizing and NBTI-aware synthesis as well as adaptive techniques such as FBB and AVS. Design-time techniques were mostly studied at the early stage of NBTI mitigation due to the limitation that they can be very time consuming and computation intensive. Recently, there have been some attempts to embed these techniques in a static time analysis engine or synthesis tool with reasonable overhead. Other circuit-level solutions are adaptive techniques: FBB, and AVS, which both hide the lengthened delays caused by NBTI degradation. According to Equation (4), delay can be reduced by either controlling V_{th} or V_{dd}. The difference between the two techniques is that FBB reduces the delay by lowering V_{th}, but AVS by lowering V_{dd} since FBB was first introduced to reduce leakage power, but AVS to reduce dynamic power.

There have been some attempts to mitigate NBTI degradation using architectural approaches. Power gating, one of the popular low-power design techniques, can improve lifetime reliability. According to Equation (3), transistor degradation is worsened as the time period that the device is under stress is longer. Power gating reduces the time period that the device is under stress and at the same time increases chances that the device experiences recovery. Self-repair as an architectural technique can improve lifetime reliability by replacing faulty components by spare components.

Traditionally, reliability problems have been regarded as the responsibility of the chip manufacturers. However, there have been recent attempts to address lifetime reliability problems at the system level. System-level techniques can reduce the cost added by circuit-level techniques, which are normally based on the worst-case degradation estimation. Traditional low-power and thermal management techniques can be successfully extended to deal with reliability problems since aging is dependent on power consumption and temperature. DVFS reduces delays caused by NBTI degradation by lowering unnecessarily high V_{dd}. DVFS, as a system-level solution, can be more efficient than FBB and AVS since they take application-level information into account. Scheduling and load-balancing techniques can also exploit application-level information. These techniques are based on different observation than DVFS. Scheduling and load-balancing techniques basically shorten stress time by assigning tasks on the device considering the level of the device's NBTI degradation. System-level self-repair is another option for enhancing the lifetime of microprocessors using either core-level or lower-level redundancies. At the architecture level, self-repair should be carefully selected and designed since it requires hardware redundancies, but it can be efficient and effective at the system level for the designs that naturally have redundancies, such as many-core processors.

With growing demand for higher performance, power density has continued to increase, which results in a thermal crisis. In addition, movement to 3D integration will make thermal problems even more serious, as power density per unit volume increases and heat dissipation becomes difficult due to vertically adjacent dies. Temperature is a major factor in accelerating NBTI degradation. Considering growing thermal problems, the performance and cost requirements of microprocessors will be considerably affected by lifetime reliability management. Various studies to enhance lifetime reliability have been conducted, but have not been sufficient to address newly introduced microprocessor design issues, such as 3D integration. Recently, studies on thermal management in 3D ICs have been published, but they do not address the long-term impact of elevated temperature on processor lifetime. More intense study is required to improve lifetime reliability in conjunction with new design issues. In addition, the growing demand for higher performance and lower power consumption may change future transistor architectures. Studies on new transistor architecture that guarantee higher lifetime reliability also have to be conducted in conjunction with system-level studies.

REFERENCES

W. Abadeer and W. Ellis. 2003. Behavior of NBTI under AC dynamic circuit conditions. In *Proceedings of the IEEE International Reliability Physics Symposium*. 17–22.

S. Aota, S. Fujii, Z. W. Jin, Y. Ito, K. Utsumi, E. Morifuji, S. Yamada, F. Matsuoka, and T. Noguchi. 2005. A new method for precise evaluation of dynamic recovery of negative bias temperature instability. In *Proceedings of the IEEE International Conference on Microelectronic Test Structures*. 197–199.

N. Ayala, J. Martin-Martinez, E. Amat, M. B. Gonzalez, P. Verheyen, R. Rodriguez, M. Nafria, X. Aymerich, and E. Simoen. 2011. NBTI related time-dependent variability of mobility and threshold voltage in pMOSFETs and their impact on circuit performance. *Microelectronic Engineering* 88, 7, 1384–1387.

M. Basoglu, M. Orshansky, and M. Erez. 2010. NBTI-aware DVFS: A new approach to saving energy and increasing processor lifetime. In *Proceedings of the ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*. 253–258.

D. R. Bild, G. E. Bok, and R. P. Dick. 2009. Minimization of NBTI performance degradation using internal node control. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'09)*. 148–153.

S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. 2003. Parameter variations and impact on circuits and microarchitecture. In *Proceedings of the ACM/IEEE Design Automation Conference*. 338–342.

K. A. Bowman, S. G. Duvall, and J. D. Meindl. 2002. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. *IEEE Journal of Solid-State Circuits* 37, 2, 183–190.

A. Calimera, E. Macii, and M. Poncino. 2009. NBTI-aware power gating for concurrent leakage and aging optimization. In *Proceedings of the International Symposium on Low Power Electronics and Design*. 127–132.

Yunus A. Cengel. 1997. Introduction to thermodynamics and heat. McGraw Hill Higher Education, Chicago, IL.

T. Chan, J. Sartori, P. Gupta, and R. Kumar. 2011. On the efficacy of NBTI mitigation techniques. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'11)*. 1–6.

T.-B. Chan, W.-T. J. Chan, and A. B. Kahng. 2013. Impact of adaptive voltage scaling on aging-aware signoff. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'13)*. 1683–1688.

A. Chaudhary and S. Mahapatra. 2013. A physical and SPICE mobility degradation analysis for NBTI. *IEEE Transactions on Electronic Devices* 60, 7, 2096–2103.

G. Chen, K. Y. Chuah, M.-F. Li, D. S. H. Chan, C. H. Ang, J. Z. Zheng, Y. Jin, and Kwong, D. L. Kwong. 2002. Dynamic NBTI of PMOS transistors and its impact on device lifetime. *IEEE Electron Device Letter* 734–736.

X. Chen, Y. Wang, Y. Cao, Y. Ma, and H. Yang. 2012. Variation-aware supply voltage assignment for simultaneous power and aging optimization. *IEEE Transactions on Very Large Scale Integration Systems* 20, 11, 2143–2147.

B. E. Deal, M. Sklar, A. S. Grove, and E. H. Snow. 1967. Characteristics of the surface-state charge (Q_{ss}) of thermally oxidized silicon. *Journal of the Electrochemical Society* 114, 3, 266–274.

M. Denais, C. Parthasarathy, G. Ribes, Y. Rey-Tauriac, N. Revil, A. Bravaix, V. Huard, and F. Perrier. 2004. On-the-fly characterization of NBTI in ultra-thin gate oxide PMOSFET's. In *Proceedings of the IEEE International Electron Devices Meeting*. 109–112.

M. Ebrahimi, F. Oboril, S. Kiamehr, and M. B. Tahoori. 2013. Aging-aware logic synthesis. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*.

S. Feng, S. Gupta, A. Ansari, and S. Mahlke. 2010. Maestro: Orchestrating lifetime reliability in chip multiprocessors. In *Proceedings of the International Conference on High Performance Embedded Architectures and Compilers*. 186–200.

R. Fernández, B. Kaczer, A. Nackaerts, S. Demuynck, R. Rodriguez, M. Nafria, and G. Groeseneken. 2006. AC NBTI studied in the 1 Hz–2 GHz range on dedicated on-chip circuits. In *Proceedings of the IEEE International Electron Devices Meeting*. 337–340.

D. Frohman-Bentchkowsky. 1971. A fully decoded 2048-bit electrically programmable FAMOS read-only memory. *IEEE Journal of Solid-State Circuits* 6, 5, 301–306.

X. Fu, T. Li, and J. Fortes. 2008. NBTI tolerant microarchitecture design in the presence of process variation. In *Proceedings of 41st IEEE/ACM International Symposium on Microarchitecture*. 399–410.

Seyab S. Hamdioui. 2010. NBTI modeling in the framework of temperature variation. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'10)*. 283–286.

J. Henkel, Bauer, L. N. Dutt, P. Gupta, and S. Nassif. 2013. Reliable on-chip systems in the nano-era: Lessons learnt and future trends. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*. 1–10.

J. Hicks, E. Bergstrom, M. Hattendorf, J. Jopling, J. Maiz, S. Pae, C. Prasad, and J. Wiedemer. 2008. 45 nm Transistor Reliability. *Intel Technology Journal* 12, 2, 131–144.

L. Huang and Q. Xu. 2010a. Energy-efficient task allocation and scheduling for multi-mode MPSoCs under lifetime reliability constraint. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'10)*. 1584–1589.

L. Huang and Q. Xu. 2010b. Characterizing the lifetime reliability of manycore processors with core-level redundancy. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 680–685.

L. Huang, F. Yuan, and Q. Xu. 2011. On task allocation and scheduling for lifetime extension of platform-based MPSoC designs. *IEEE Transactions on Parallel and Distributed Systems* 22, 12, 2088–2099.

V. Huard and M. Denais. 2004. Hole trapping effect on methodology for DC and AC negative bias temperature instability measurements in PMOS transistors. In *Proceedings of the International Reliability Physics Symposium*. 40–45.

V. Huard, M. Denais, and C. Parthasarathy. 2006. NBTI degradation: From physical mechanisms to modeling. *Microelectronics Reliability* 46, 1, 1–23.

V. Huard, C. Parthasarathy, N. Rallet, C. Guerin, M. Mammase, D. Barge, and C. Ouvrard. 2007. New characterization and modeling approach for NBTI degradation from transistor to product level. In *Proceedings of the IEEE International Electron Devices Meeting (IEDM)*. 797–800.

T. Jin and S. Wang. 2012. Aging-aware instruction cache design by duty cycle balancing. In *Proceedings of the Computer Society Annual Symposium on VLSI*. 195–200.

K. Kang, H. Kufluoglu, M. A. Alain, and K. Roy. 2006. Efficient transistor-level sizing technique under temporal performance degradation due to NBTI. In *Proceedings of the International Conference on Computer Design (ICCD'06)*.

K. Kang, S. Gangwal, S. Park, and K. Roy. 2008. NBTI induced performance degradation in logic and memory circuits: how effectively can we approach a reliability solution? In *Proceedings of the Asia and South Pacific Design Automation Conference*. 726–731.

G. Karakonstantis, C. Augustine, and K. Roy. 2010. A self-consistent model to estimate NBTI degradation and a comprehensive on-line system lifetime enhancement technique. In *Proceedings of the IEEE 16th International On-Line Testing Symposium (IOLTS)*. 3–8.

U. R. Karpuzcu, B. Greskamp, and J. Torrellas. 2009. The BubbleWrap many-core: Popping cores for sequential acceleration. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO'09)*. 447–458.

J. Keane, T.-H. Kim, and C. H. Kim. 2010. An on-chip NBTI sensor for measuring PMOS threshold voltage degradation. *IEEE Transactions on Very Large Scale Integration Systems* 18, 6, 947–956.

J. Keane, D. Persaud, and. C. H. Kim. 2009. An all-in-one silicon Odometer for separately monitoring HCI, BTI, and TDDB. In *Proceedings of the IEEE Symposium on VLSI Circuits*. 108–109.

M. B. Ketchen, M. Bhushan, and R. Bolam. 2007. Ring oscillator based test structure for NBTI analysis. In *Proceedings of the IEEE International Conference on Microelectronic Test Structures*. 42–47.

S. Khan, N. Z. Haron, S. Hamdioui, and F. Catthoor. 2011. NBTI monitoring and design for reliability in nanoscale circuits. In *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. 68–76.

T.-H. Kim, R. Persaud, and C. H. Kim. 2008. Silicon odometer: An on-chip reliability monitor for measuring frequency degradation of digital circuits. *IEEE Journal of Solid-State Circuits* 43, 4, 874–880.

P. Ko, J. Huang, Z. Liu, and C. Hu. 1993. BSIM3 for analog and digital circuit simulation. In *Proceedings of the IEEE Symposium on VLSI Technology CAD*. 400–429.

S. K. Krishnappa and H. Mahmoodi. 2011. Comparative BTI reliability analysis of SRAM cell designs in nano-scale CMOS technology. In *Proceedings of the IEEE International Symposium on Quality Electronic Design*. 1–6.

S. Kumar, K. H. Kim, and S. S. Sapatnekar. 2006. Impact of NBTI on SRAM read stability and design for reliability. In *Proceedings of the IEEE International Symposium on Quality Electronic Design*. 210–218.

S. V. Kumar, C. H. Kim, and S. S. Sapatnekar. 2007. NBTI-aware synthesis of digital circuits. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*. 370–375.

S. V. Kumar, C. H. Kim, and S. S. Sapatnekar. 2011. Adaptive techniques for overcoming performance degradation due to aging in CMOS circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 19, 4, 603–614.

G. La Rosa, F. Guarin, S. Rauch, A. Acovic, J. Lukaitis, and E. Crabbe. 1997. NBTI-channel hot carrier effects in PMOSFETs in advanced CMOS technologies. In *Proceedings of the IEEE International Reliability Physics Symposium*. 282–286.

L. Lai, V. Chandra, R. Aitken, and P. Gupta. 2014. BTI-Gater: An aging-resilient clock gating methodology. *IEEE Journal on Emerging and Selected Topics in Circuits and System* 4, 2, 180–189.

Y. Lee and T. Kim. 2011. A fine-grained technique of NBTI-aware voltage scaling and body biasing for standard cell based designs. In *Proceedings of the Asia and South Pacific Design Automation Conference*. 603–608.

L. Li, Y. Zhang, and J. Yang. 2011. Proactive recovery for BTI in high-k SRAM cells. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'11)*. 1–6.

Z.-H. Liu, C. Hu, J.-H. Huang, T.-Y. Chan, M.-C. Jeng, P. K. Ko, and Y. C. Cheng. 1993. Threshold voltage model for deep submicrometer MOSFETs. *IEEE Transactions on Electron Devices* 40, 1, 86–95.

Y. Lu, L. Shang, H. Zhou, F. Yang, and X. Zeng. 2009. Statistical reliability analysis under process variation and aging effects. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*. 514–519.

H. Masuda, S. Ohkawa, A. Kurokawa, and M. Aoki. 2005. Challenge: Variability characterization and modeling for 65-nm to 90-nm processes. In *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*. 593–599.

E. Mintarno, J. Sckaf, R. Zheng, J. B. Velamala, Y. Cao, S. Boyd, R. W. Dutton, and S. Mitra. 2011. Self-tuning for maximized lifetime energy-efficiency in the presence of circuit aging. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30, 5, 760–773.

H. Mostafa, M. Anis, and M. Elmasry. 2011. Adaptive Body Bias for Reducing the Impacts of NBTI and Process Variations on 6T SRAM Cells. *IEEE Transactions on Circuits and Systems I* 58, 12, 2859–2871.

H. Mostafa, M. Anis, and M. Elmasry. 2012. NBTI and process variations compensation circuits using adaptive body bias. *IEEE Transactions on Semiconductor Manufacturing* 25, 3, 460–467.

S. Narendra, A. Keshavarzi, B. A. Bloechel, S. Borkar, and V. De. 2003. Forward body bias for microprocessors in 130-nm technology generation and beyond. *IEEE Journal of Solid State Circuits* 38, 5, 696–701.

A. Neugroschel, C. T. Sah, K. M. Han, M. S. Carroll, T. Nishida, J. T. Kavalieros, and Y. Lu. 1995. Direct-current measurement of oxide and interface traps on oxidized silicon. *IEEE Transactions on Electron Devices* 42, 9, 1657–1662.

F. Oboril and M. B. Tahoori. 2012a. Reducing wearout in embedded processors using proactive fine-grain dynamic runtime adaptation. In *Proceedings of the IEEE European Test Symposium (ETS)*. 1–6.

F. Oboril and M. B. Tahoori. 2012b. ExtraTime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level reducing. In *Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN)*.

F. Oboril, F. Firouzi, S. Kiamehr, and M. B. Tahoori. 2013. Negative bias temperature instability-aware instruction scheduling: A cross-layer approach. *Journal of Low Power Electronics*, 9, 4, 389–402.

S. Ogawa and N. Shiono. 1995. Generalized diffusion-reaction model for the low-field charge-buildup instability at the Si-SiO$_2$ interface. *Physical Review B* 51, 7, 4218–4230.

S. Pae, M. Agostinelli, M. Brazier, G. Chau, G. Dewey, Y. Ghani, M. Hattendorf, J. Hicks, J. Kavalieros, M. Kuhn, J. Maiz, M. Metz, K. Mistry, C. Prasad, S. Ramey, A. Roskowski, J. Sandford, C. Thomas, J. Thomas, C. Wiegand, and J. Wiedemer. 2008. BTI reliability of 45 nm high-k + metal-gate process technology. *IEEE International Reliability Physics Symposium* 352–357.

B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy. 2005. Impact of NBTI on the temporal performance degradation of digital circuits. *IEEE Electron Device Letter* 560–562.

B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy. 2006. Temporal performance degradation under NBTI: Estimation and design for improved reliability of nanoscale circuits. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'06)*.

A. Pushkarna and H. Mahmoodi. 2010. Reliability analysis of power gated SRAM under combined effects of NBTI and PBTI in Nano-Scale CMOS. In *Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI)*.

Z. Qi and M. R. Stan. 2008. NBTI resilient circuits using adaptive body biasing. In *Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI)*. 285–290.

S. Rangan, N. Mielke, and E. C. C. Yeh. 2003. Universal recovery behavior of negative bias temperature instability. In *Proceedings of the IEEE International Electron Devices Meeting*. 341–344.

B. F. Romanescu and D. J. Sorin. 2008. Core cannibalization architecture: Improving lifetime chip performance for multicore processors in the presence of hard faults. In *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques (PACT)*. 1–10.

S. Roy and D. Z. Pan. 2014. Reliability aware gate sizing combating NBTI and oxide breakdown. In *Proceedings of the International Conference on VLSI Design and International Conference on Embedded Systems*. 38–43.

N. Sa, J. F. Kand, H. Yang, X. Y. Liu, Y. D. He, R. Q. Han, C. Ren, H. Y. Yu, D. S. H. Chan, and D. L. Kwong. 2002. Mechanism of positive-bias temperature instability in sub-1-nm TaN/HfN/HfO$_2$ gate stack with low preexisting traps. *IEEE Electron Device Letters* 26, 9, 610–612.

T. Sakurai and A. R. Newton. 1990. Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas. *IEEE Journal of Solid-State Circuits* 25, 2, 584–594.

C. Shen, C. E. Li, M.-F. Foo, T. Yang, D. M. Huang, A. Yap, G. S. Samudra, and Y.-C. Yeo. 2006. Characterization and physical origin of fast V$_{th}$ transient in NBTI of pMOSFETs with SiON dielectrics. In *Proceedings of the IEEE International Electron Devices Meeting*. 1–4.

T. Siddiqua and S. Gurumurthi. 2009. NBTI-aware dynamic instruction scheduling. In *Proceedings of the IEEE Workshop on Silicon Errors in Logic–System Effects*.

T. Siddiqua and S. Gurumurthi. 2010. A multi-level approach to reduce the impact of NBTI on processor functional units. In *Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI)*. 67–72.

T. Siddiqua, S. Gurumurthi, and M. R. Stan. 2011. Modeling and analyzing NBTI in the presence of process variation. In *Proceedings of the Quality of Electronic Design (ISQED)*. 1–8.

P. Singh, E. Karl, D. Blaauw, and D. Sylvester. 2012. Compact degradation sensors for monitoring NBTI and oxide degradation. *IEEE Transactions on Very Large Scale Integration Systems* 20, 9, 1645–1655.

J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers. 2004. The case for lifetime reliability-aware micorprocessors. In *Proceedings of the International Symposium on Computer Architecture (ISCA'04)*.

J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers. 2005a. Lifetime reliability: Toward an architectural solution. *IEEE Micro* 25, 3, 70–80.

J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers. 2005b. Exploiting structural duplication for lifetime reliability enhancement. In *Proceedings of the International Symposium on Computer Architecture (ISCA'05)*.

N. Strikos. 2013. Enhancing Lifetime Reliability Of Chip Multiprocessors Through 3D Resource Sharing. Master's thesis. University of California, San Diego.

J. Sun, A. Kodi, A. Louri, and J. M. Wang. 2009. NBTI aware workload balancing in multi-core systems. In *Proceedings of the Quality of Electronic Design (ISQED)*. 833–838.

A. Tiwari and J. Torrellas. 2008. Facelift: Hiding and slowing down aging in multicores. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*. 129–140.

K. Uwasawa, T. Yamamoto, and T. Mogami. 1995. A new degradation mode of scaled p+ polysilicon gate pMOSFETs induced by bias temperature (BT) instability. In *Proceedings of the IEEE International Electron Devices Meeting*. 871–874.

R. Vattikonda, W. Wang, and Y. Cao. 2006. Modeling and minimization of PMOS NBTI effect for robust nanometer design. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'06)*. 1047–1052.

I. Wagner and V. Bertacco. 2008. Reversi: Post-silicon validation system for modern micro-processors. In *Proceedings of the International Conference on Computer Design (ICCD'08)*.

W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao. 2007a. The impact of NBTI on the performance of combinational and sequential circuits. In *Proceedings of the ACM/IEEE Design Automation Conference*. 364–369.

Y. Wang, H. Luo, K. He, R. Luo, H. Yang, and Y. Xie. 2007b. Temperature-aware NBTI modeling and the impact of input vector control on performance degradation. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE'07)*. 546–551.

Y. Wang, X. Chen, W. Wang, V. Balakrishnan, Y. Cao, Y. Xie, and H. Yang. 2009. On the efficacy of input vector control to mitigate NBTI effects and leakage power. In *Proceedings of the International Symposium on Quality of Electronic Design*. 19–26.

Y. Wang, S. Cotofana, and F. Liang. 2011. A unified aging model of NBTI and HCI degradation towards lifetime reliability management for nanoscale MOSFET. In *Proceedings of the International Symposium on Nanoscale Architectures (NANOARCH)*. 175–180.

M. White. 2008. Microelectronics reliability: physics-of-failure based modeling and lifetime evaluation. Technical Report. Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA.

H. Yang, W. Hwang, and C.-T. Chuang. 2011. Impacts of NBTI/PBTI and contact resistance on power-gated SRAM with high-k metal-gate devices. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 19, 7, 1192–1204.

X. Yang and S. Saluja. 2007. Combating NBTI degradation via gate sizing. In *Proceedings of the Quality of Electronic Design (ISQED)*. 47–52.

S. Zafar, B. H. Lee, J. Stathis, and A. Callegari. 2004. A model for negative bias temperature instability (NBTI) in oxide and high k pFETs. In *Proceedings of the Symposium on VLSI Technology*. 208–209.

S. Zafar, Y. H. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stahis, A. Callegari, and M. Chudzik. 2006. A comparative study of NBTI and PBTI (Charge Trapping) in SiO_2/HfO_2 Stacks with FUSI, TiN, Re Gates. In *Proceedings of the Symposium on VLSI Technology*. 23–25.

L. Zhang and R. P. Dick. 2009. Scheduled voltage scaling for increasing lifetime in the presence of NBTI. In *Proceedings of the Asia and South Pacific Design Automation Conference*. 492–497.

K. Zhao, J. H. Stathis, B. P. Linder, E. Cartier, and A. Kerber. 2011. PBTI under dynamic stress: From a single defect point of view. *IEEE International Reliability Physics Symposium* 4A.3.1–4A.3.9.