SRAM Stability Analysis and Performance–Reliability Tradeoff for Different Cache Configurations

Rui Zhang, Taizhi Liu, Kexin Yang, Chang-Chih Chen, and Linda Milor, *Senior Member, IEEE*

***Abstract*— Bias temperature instability (BTI), hot carrier injection (HCI), gate–oxide time-dependent dielectric breakdown (GTDDB), and random telegraph noise (RTN) degrade the stability of the deeply scaled transistors and the overall circuit reliability. These front-end wearout mechanisms are especially acute in the static random access memory (SRAM) cells of first-level (L1) caches, which are crucial for the performance of microprocessors due to frequent accesses. This article presents a methodology to analyze cache reliability degradation due to the combined effect of BTI, HCI, GTDDB, and RTN for different cache configurations, including variations due to associativity, cache line size, cache size, and the error-correcting codes (ECCs). Time-zero variability due to process and environmental parameters are also considered. First, we analyze how each wearout mechanism affects reliability degradation. Then we analyze the relationship between reliability (probability of failure) and performance (hit rate) of the L1 cache within a LEON3 microprocessor, while the LEON3 is running a set of benchmarks, which determine cell array activity, characterized by the duty cycle, toggle rate, temperature, and supply voltage distributions of cells. Insights on the performance–reliability tradeoff are provided for cache designers.**

***Index Terms*— Bias temperature instability (BTI), cache configurations, error-correcting codes (ECCs), gate–oxide breakdown (GTDDB), hot carrier injection (HCI), LEON3 microprocessor, performance–reliability tradeoff, random telegraph noise (RTN), time-dependent dielectric breakdown, wearout.**

I. Introduction

STATIC random access memories (SRAMs) are the dominant part of systems-on-chips (SoCs). They consume half or more than half of the die area and most of the transistors in modern microprocessors. Hence, it is important to analyze the reliability of SRAMs. This article focuses on the first-level (L1) data cache. This block may not be the worst block from the reliability perspective, as mentioned in [1]–[3], which suggests the greater importance of the instruction caches (I-Caches) and register files, but it is analyzed in detail to illustrate the methodology. Cache efficiency is also critical for system performance. Many prior works have focused on the cache architecture needed to obtain higher cache efficiency [4]–[6]. However, it is not known how cache reliability is affected when a higher performance is achieved. In this article, we present a methodology to estimate cache reliability and apply this methodology to investigate the reliability [probability of failure (PF)] of the L1 data cache in a typical SoC microprocessor for different design configurations, by looking at associativity, cache line size, cache size, and error-correcting codes (ECCs). By analyzing the reliability and performances of different cache configurations, we provide insights into how to achieve the best performance–reliability tradeoff in cache system design.

Advanced SRAM design is accompanied with the development of technology. Although smaller technology nodes bring various benefits, like higher device density and lower power consumption, they also pose significant reliability challenges. Deeply scaled CMOS devices, such as Fin Field-Effect Transistors (FinFETs), have a high sensitivity to process parameter variability and front-end wearout mechanisms, such as bias temperature instability (BTI), hot carrier injection (HCI), gate–oxide breakdown (GTDDB), and random telegraph noise (RTN). Variability and wearout not only make transistors unreliable for low-voltage operation, but also lead to earlier functional failures of circuits. This concern is for all computing devices, ranging from server processors, where lifetime is a critical requirement, to mobile devices, where the market share strongly depends on reliability.

This article takes into account performance degradation due to BTI, HCI, GTDDB, and RTN. These wearout mechanisms degrade SRAM cell performances, including read/write/hold static noise margins (SNMs), minimum voltage for state retention, read delay and power, write delay and power, and the leakage power during hold. The degradation of these performances is random, involving within-die and between-die process parameter variations, in addition to the degradation caused by wearout.

The SRAM is composed of cells, but to determine SRAM lifetime, the stress distribution among the cells must be taken

Manuscript received April 14, 2019; revised August 14, 2019 and October 18, 2019; accepted November 12, 2019. Date of publication January 21, 2020; date of current version February 25, 2020. This work was supported by the National Science Foundation under Award 1700914. *(Corresponding author: Linda Milor.)*

R. Zhang and L. Milor are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ruizhang348@gatech.edu; linda.milor@ece.gatech.edu).

T. Liu was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. He is now with Cadence Design Systems, San Jose, CA 95134 USA (e-mail: taizhi@cadence.com).

K. Yang was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. He is now with Synopsys, Mountain View, CA 94043 USA (e-mail: kexin.yang@synopsys.com).

C.-C. Chen was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. He is now with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: changc@microsoft.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2019.2956923

into account, which depends on the use scenario of the microprocessor. In this article, SRAM performances [hit rate (HR)] and lifetime are evaluated for the full SRAM when the microprocessor (containing the SRAM) is running realistic workloads. The workloads are determined by running a set of benchmarks on an emulation system, which determines the duty cycle and toggle rate distributions for the SRAM cells in the SRAM array. These parameters are then used to determine the stress caused by each wearout mechanism in each cell. The cell lifetime distributions are then computed and combined to determine the statistical lifetime distribution and failure probability of the full SRAM.

This work is new in the following ways.

1) A methodology that incorporates degradation due to BTI, HCI, RTN, and GTDDB in a unified way is presented. Shifts due to wearout are combined with within-die and between-die process parameters for a complete parametric description of degradation.
2) The methodology is demonstrated with application to the analysis of cache reliability for different cache configurations considering associativity, cache line size, and cache size.
3) The impact of variations in device parameters (process/electrical) is combined with wearout parameters in Monte Carlo (MC) simulations to find the failure probability of the SRAM cells based on when various performance metrics degrade beyond design limitations.
4) The work uses the stress profile of the microprocessor to analyze the cache reliability which is determined with a field-programmable gate array (FPGA)-based aging assessment framework to determine the stress profile distributions of the cells in the SRAM.

The remainder of this article is organized as follows. Section II summarizes the wearout models and the SRAM cell structure studied in this article. Section III describes the steps for activity extraction and shows an example of activity distributions. It also presents the aging assessment framework. Section IV uses the aging assessment framework to evaluate SRAM cell and cache degradation. Section V analyzes how various configurations affect cache performance and reliability. Section VI concludes this article.

II. Background and Prior Art

Our evaluation process starts from physical models and propagates them to the cache level, where they cause a shift in performance metrics. In this section, we first introduce the models for each wearout mechanism and how they affect the device parameters. Then, we discuss how the mechanisms degrade SRAM cell performances under various stress conditions. Finally, we summarize prior work about how wearout mechanisms impact SRAM reliability.

A. Wearout Mechanisms (BTI, HCI, GTDDB, and RTN)

1) BTI Model: BTI is caused by trap generation and degeneration at the interface and in the bulk of gate dielectric materials. Studies of BTI have been conducted at the transistor and gate level with the classical reaction–diffusion (R–D) model and the atomistic trap-based model [7]–[10]. Consensus between the two models has not been achieved yet. The classical R–D model has a lower computational and memory requirement, whereas the atomistic model has a higher resolution in the nanosecond range and is challenging to use for longer stress simulations [11]. In this article, since the simulated time ranges from 1 to 10^8 s, an enhanced R–D model is adopted to emulate the high-k dielectric charge evolution with time, as described in [12]–[14]. Because of the less importance of nFET BTI (PBTI), only the pFET BTI (NBTI) is taken into account [15], [16].

Three uncorrelated contributions from the generation of the interface trap density (ΔN_{IT}), hole trapping in preexisting sites (ΔN_{HT}), and the generation of new bulk insulator (ΔN_{OT}) traps are applied for trap density evolution [12]. The model also incorporates the stress-recovery phenomenon, duty factor, and the effect of temperature and operating frequency.

For BTI, the time range and the ratios of stress and recovery are quite important for a clear prediction of degradation. There is a complete solution for stress and recovery for each part under short time durations, and a simplified solution for each part under long-term dc stress [12]. Considering the long times in this article, we combine the long-term dc stress model with a duty factor equation to calculate the overall BTI degradation. The traps' shift due to BTI is predicted as

$$\Delta N_{IT} = A(V_G - V_{T0} - \Delta V_T)^{\Gamma_{IT}} e^{-\frac{E_{AIT}}{kT}} t^{\frac{1}{6}} \tag{1a}$$

$$E_{AIT} = \frac{2}{3}(E_{Akf} - E_{Akr}) + \frac{E_{ADH2}}{6} \tag{1b}$$

$$\Delta N_{HT} = B(V_G - V_{T0} - \Delta V_T)^{\Gamma_{HT}} e^{-\frac{E_{AHT}}{kT}} \tag{2}$$

$$\Delta N_{OT} = C\left(1 - e^{\left(-\left(\frac{t}{n}\right)^{\beta_{OT}}\right)}\right) \tag{3a}$$

$$n = \eta(V_G - V_{T0} - \Delta V_T)^{\frac{\Gamma_{OT}}{\beta_{OT}}} e^{-\frac{E_{AOT}}{kT\beta_{OT}}} \tag{3b}$$

where A, B, C, Γ_{IT}, E_{AIT}, E_{Akf}, E_{Akr}, E_{ADH2}, Γ_{HT}, E_{AHT}, η, Γ_{OT}, β_{OT}, and E_{AOT} are constants. V_G is the stress voltage, V_{T0} is the initial threshold voltage, ΔV_T is the shift of the threshold voltage, t is the stress time, k is Boltzmann's constant, and T is temperature.

We use constants adopted from [12] to calculate NBTI degradation for high-k devices with $EOT = 0.7$ nm. A, B, and C are scaled simultaneously to meet the assumption that $\Delta V_{T,dc,Mean}$ is 100 mV after ten years of dc stress [17].

Since SRAM cells experience frequent *Read/Write* operations, ac waveforms are necessary to predict real NBTI degradation under different duty cycles. A universal relaxation model for the effect of duty cycle (D) has been proposed [18] and validated [19] for FinFETs and planar devices in 16- and 20-nm technology, respectively. The equivalent shift due to traps (impacting the threshold voltage, etc.) is the product of the shift under dc stress and the recovery fraction, $r(D)$. In general, $r(D)$ is a function of operating frequency, temperature, and stress voltage [18]. We adopt a simplified expression for $r(D)$ due to the lack of experimental data on the impact of frequency, temperature, and stress voltage

$$r(D) = \frac{1}{1 + B_{DF}\left(\frac{1}{D} - 1\right)^{\beta_{DF}}} \tag{4}$$

where B_{DF} is a scaling factor and β_{DF} is a dispersive shape factor [20]. The accuracy of the simplified recovery fraction model for experimental data (under various stress conditions) is validated in [18].

2) HCI Model: HCI happens when there are hot carriers flowing through the channel. Energy transferred from hot carriers to the lattice helps generate interface states or bulk defects [21], [22]. HCI is caused by both interfaces and oxide bulk trapped charges [23]. It is found that the interface charge is not recoverable, however the bulk charge is partially recoverable, and the highest HCI degradation in FinFETs appears at $V_G = V_D$ [23]. It has been found that degradation caused by HCI is primarily due to interface traps and is not recoverable [24], and modeling of interface charge matches well with experimental data [25], [26]. Hence, we adopt the analytical model in [25] to describe how the threshold voltage related to HCI evolves with time. The interface charge is the only contributor and is not recoverable. HCI in pFETs is three times higher than that in nFETs under the same stress conditions [27].

The interface trap degradation due to HCI during the time under stress varies with FinFET dimension, stress voltage, and temperature [25], [28]. It is modeled as

$$\Delta N_{IT} = Dt^n (1/L_g)^b * \exp(c_1 V_{ds}) \times \exp(-c_2(V_{ds} - V_{gs}))\exp(E_{a,\mathrm{HCI}}/kT) \quad (5)$$

where D, n, b, c_1, and c_2 are constants. L_g is the gate length and V_{ds} and V_{gs} are the drain–source voltage and the gate–source voltage, respectively. $E_{a,\mathrm{HCI}}$ is the activation energy.

HCI occurs when highly energized (hot) carriers flow from the drain to the source. Therefore, in the SRAM cell structure, HCI happens when a transistor is ON and is conducting current. Therefore, HCI happens when the stored data is being flipped. It is necessary to figure out the equivalent stress time during a transition, so that (5) can be evaluated for fixed values of $V_{ds} = V_{DD}$ and $V_{gs} = V_{DD}$. A method described in [29] is used to obtain the equivalent stress time for each transition. The equivalent times are summed to determine the total equivalent time interval for a transition. The values of equivalent times are combined with the transition rate computed in Section III.

It has been checked that HCI does not affect the threshold voltage by more than 2 mV after ten years of stress at a constant operating frequency of 250 MHz using the model parameters in [25] and [28].

3) GTDDB Model: With the increase in stress time, traps are accumulated in the oxide layer. If the traps do not overlap from the gate to the channel, the device gate leakage current is not affected. GTDDB occurs when the conducting paths are formed in the gate dielectric. According to its severity, GTDDB is divided into soft breakdown (SBD) and hard breakdown (HBD). HBD has leakage currents that are several orders of magnitude larger than SBD. For ultrathin dielectrics, conversion between the two becomes more unstable. Therefore, we use a statistical model for the GTDDB failure time distribution. Time to HBD is the sum of time to SBD (T_{BD}) and the time gap between SBD and HBD (T_{PBD}), where both T_{BD} and T_{PBD} follow Weibull distributions [30].

Specifically, the breakdown (BD) of ultrathin gate–oxides consists of the first BD and the progressive BD (PBD) phases. Thus, the overall time to oxide failure is the sum of the time to first BD (T_{BD}) and the duration of PBD (T_{PBD}) [31]. Before the first BD, the leakage current is negligible (set as $I_{\mathrm{BD}}(t) = 1.0$ nA, $0 \le t \le T_{\mathrm{BD}}$). When BD enters PBD, the current starts to rise. At the end of PBD, the leakage current reaches a saturation level which is taken as HBD. Here we set the HBD current as 1 μA ($I_{\mathrm{BD}}(T_{\mathrm{BD}} + T_{\mathrm{PBD}})$) for FinFETs [32]. There are linear and exponential growth models for this process. The exponential model is applied in this work [33], [34]

$$I_{\mathrm{BD}}(t) = I_{\mathrm{BD}}(T_{\mathrm{BD}})\exp\left(\frac{t - T_{\mathrm{BD}}}{\tau_{\mathrm{BD}}}\right), \quad t_{\mathrm{BD}} \le t \le T_{\mathrm{BD}} + T_{\mathrm{PBD}} \quad (6a)$$

$$\tau_{\mathrm{BD}} = \frac{T_{\mathrm{PBD}}}{3 \cdot \ln(10)} \quad (6b)$$

where T_{PBD} is a Weibull distribution.

The overall time to failure is $T_{\mathrm{FAIL}} = T_{\mathrm{BD}} + T_{\mathrm{PBD}}$. Both T_{BD} and T_{PBD} follow Weibull distributions. For small-area devices in an SRAM cell, T_{PBD} is much smaller than T_{BD}. It is reasonable to assume that T_{PBD} follows a Weibull distribution with a scale parameter of 10^6 s (η_{PBD}) and a shape parameter of 1 (β_{PBD}) [31]. T_{BD} has a shape parameter of 1.08 (β_{BD}) and a scale parameter as follows:

$$\eta_{\mathrm{BD}} = E V_G^{-n_{\mathrm{BD}}} \left(\frac{1}{A_{\mathrm{eff}}}\right)^{\frac{1}{\beta_{\mathrm{BD}}}} \exp\left(\frac{E_{a,\mathrm{BD}}}{k_b T}\right) \quad (7)$$

where E and n_{BD} are constants, A_{eff} is the effective gate area, and $E_{a,\mathrm{BD}}$ is the activation energy.

Under these assumptions, the cumulative distribution function (CDF) of T_{fail} is [30]

$$F_{\mathrm{FAIL}}(t) = 1 - \exp\left\{-\left[\frac{\left(\frac{t}{T_F}\right)^{\beta_F}\left(\frac{t}{T_{\mathrm{BD}}}\right)^{\beta_{\mathrm{BD}}}}{\left(\frac{t}{T_F}\right)^{\beta_F} + \left(\frac{t}{T_{\mathrm{BD}}}\right)^{\beta_{\mathrm{BD}}}}\right]\right\} \quad (8a)$$

$$\beta_F = \beta_{\mathrm{BD}} + \beta_{\mathrm{PBD}} \quad (8b)$$

$$T_F = \left(\frac{T_{\mathrm{BD}}^{\beta_{\mathrm{BD}}} T_{\mathrm{PBD}}^{\beta_{\mathrm{PBD}}}}{K}\right)^{\frac{1}{\beta_F}} \quad (8c)$$

$$K = \frac{\beta_{\mathrm{BD}}\beta_{\mathrm{PBD}}}{\beta_{\mathrm{BD}} + \beta_{\mathrm{PBD}}} B(\beta_{\mathrm{BD}}, \beta_{\mathrm{PBD}}) \quad (8d)$$

where $B(\cdot, \cdot)$ is the beta function.

In the light of models in (6)–(8), it is convenient to calculate the distribution of gate leakage current under specific stress conditions (such as time, temperature, and stress voltage). Moreover, ratios of leakage current flowing from the gate to the drain and the source depend on the position of percolation paths. In our simulations, it is assumed that the location of the percolation path is uniformly distributed within the device channel [35]. Since TDDB is less of a concern for pFETs [15], we only consider nFET GTDDB in this article.

Our models include only the leakage current impact of GTDDB. However, it is now known that BTI and GTDDB accelerate each other [36]. Hence, in the future, the defect generation caused by GTDDB can be included in the defect density models that are used to determine the device parameter degradation. Similarly, defects from BTI could serve to accelerate GTDDB.

4) RTN Model: The trapping and de-trapping of channel charges contribute to RTN by introducing large variations in the interface trap density (ΔN_{IT}). Variations in ΔN_{IT} cause variations in various device parameters, such as the threshold voltage (V_{th}). RTN leads to the malfunction of circuits when ΔN_{IT} is large. The distribution of ΔN_{IT} has been observed to be a lognormal distribution for deeply scaled devices [37], [38]. We similarly model RTN as a lognormal distribution.

RTN is affected by device dimensions, temperature, and interface charge density [37], [39]. RTN introduces additional variations in the interface trap density (ΔN_{IT}). However, since the fluctuation is temporary, it does not affect the accumulated ΔN_{IT} induced by BTI and HCI. ΔN_{IT} induced by RTN is modeled with a lognormal distribution, with mean and standard deviation as follows [37], [39]:

$$\mu_{\ln(\Delta N_{IT})} = F + \ln\left(\frac{1}{W_{\text{eff}} L_{\text{eff}}}\right) \tag{9a}$$

$$\sigma_{\ln(\Delta N_{IT})} = \frac{qG(\Delta V_{IT} + V_{T0})}{n_{\text{RTN}} k_b T} + \ln\left[\frac{HC_{\text{ox}}}{q(W_{\text{eff}} L_{\text{eff}})^{m_{\text{RTN}}}}\right] \tag{9b}$$

where F, G, H, n_{RTN}, and m_{RTN} are constants. W_{eff} and L_{eff} are the effective gate width and length, respectively. q is the elementary charge. $C_{\text{ox}} = \varepsilon_{\text{SiO2}}/EOT$ is the gate capacitance. $\varepsilon_{\text{SiO2}}$ is the dielectric constant of SiO_2.

5) Impact on Device Parameters: In this article, BTI is modeled as affecting the interface traps of the pFETs, hole trapping in preexisting sites, and the generation of new bulk insulator traps. HCI affects interface traps of the n/pFETs. RTN introduces extra variations in the number of traps, while GTDDB leads to gate leakage current in the nFETs.

It is verified in [40] that performance evaluation considering solely the impact on V_T draws over optimistic conclusions. To obtain a convincing and persuasive result, it is necessary to include a more comprehensive model of the impact on the device parameters. Therefore, a shift of charge density in a transistor, caused by BTI, HCI, and RTN, leads to a shift of threshold voltage, carrier mobility (μ), subthreshold slope (SS), and gate–drain capacitance (C_{gd}) [44]. The GTDDB-induced gate leakage current consists of leakage from the gate to the drain and leakage from the gate to the source. Leakage currents are implemented with Verilog-A models.

Uncontrollable factors in manufacturing cause time-zero variability of the threshold voltage. Wearout causes both the mean and standard deviation to shift with time under stress.

The overall shift of the threshold voltage due to NBTI and HCI is modeled as a normal distribution with mean, μ_{V_T}, and variance $\sigma^2_{V_{T0}}$ [41]

$$\mu_{V_T} = \frac{q(\Delta N_{IT} + \Delta N_{HT} + \Delta N_{OT})}{C_{\text{ox}}} + V_{T_0} \tag{10a}$$

$$\sigma^2_{V_T} = \left(1 + \frac{\Delta V_T}{100\ \text{mV}}\right)\sigma^2_{V_{T0}} \tag{10b}$$

where the trap shifts are caused by BTI and HCI. V_{T_0} and $\sigma^2_{V_{T0}}$ are the time-zero mean and variance, respectively. They depend on the specific manufacturing process. We assume a well-controlled process by setting the square root of the variance to be 5% of V_{T0} [42].

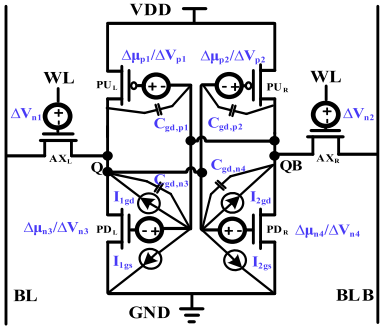

Fig. 1. Typical 6T SRAM cell with degradation and variability parameters marked.

The threshold voltage shift (ΔV_{IT}) caused by RTN is obtained from the lognormal distribution with a mean and standard deviation:

$$\mu_{\ln(\Delta V_{IT})} = \frac{q}{C_{\text{ox}}}\mu_{\ln(\Delta N_{IT})} \tag{11a}$$

$$\sigma_{\ln(\Delta V_{IT})} = \frac{q}{C_{\text{ox}}}\sigma_{\ln(\Delta N_{IT})}. \tag{11b}$$

The overall threshold voltage shift distribution is sum of the normal distribution described in (10) and the lognormal distribution shown in (11).

These front-end wearout mechanisms induce not only ΔN_{IT}, but also ΔN_{HT} and ΔN_{OT}. Only shallow ΔN_{IT} impacts μ, SS, and C_{gd}. The relationship between effective carrier mobility and the threshold voltage shift induced by interface traps (through Coulomb scattering) is described by [25]

$$\mu_{\text{eff}} = \frac{\mu_{\text{eff,ini}}}{1 + \alpha_1 \Delta V_{IT}} \tag{12}$$

where $\mu_{\text{eff,ini}}$ is the effective carrier mobility initially, ΔV_{IT} is the threshold voltage shift due to interface traps, and α_1 is a constant.

It is found in [40] that SS is sensitive to interface trap capacitance (C_{IT}) and ΔV_{IT}. In this article, we use the 14-nm predictive technology model (PTM) for simulation [43]. We have checked how C_{IT} and ΔV_{IT} affect SS for devices with various gate lengths and temperatures. Neither C_{IT} nor ΔV_{IT} affect SS in an obvious way. Therefore, the shift of SS is not considered. It can be easily added for other device models that are sensitive to C_{IT} and ΔV_{IT}.

C_{gd} degradation due to ΔN_{IT} is characterized with TCAD simulations, incorporating real device materials and dimensions. Since the influence of ΔN_{IT} on C_{gd} varies for different values of L_g, it is also included in the characterization. A generalized expression of this relation is

$$\begin{aligned}\Delta C_{gd} = {} & p_0 + p_1 \Delta N_{IT} + p_2 L_g + p_3 \Delta N_{IT}^2 + p_4 \Delta N_{IT} L_g \\ & + p_5 L_g^2 + p_6 \Delta N_{IT}^3 + p_7 \Delta N_{IT}^2 L_g + p_8 \Delta N_{IT} L_g\end{aligned} \tag{13}$$

where p_0–p_8 are constants. The units of ΔN_{IT} and L_g are cm^{-2} and nanometer, respectively.

B. SRAM Cell

Each cached bit is implemented with an SRAM cell consisting of six transistors (6T) as shown in Fig. 1. The cell is implemented with the PTM for the 14-nm technology node [43]. The cell configuration is designed with a fin number ratio of 1:1:2 (PU:AX:PD). The degradation and variability parameters considered in this article are marked. The labeled transistors form an inverter loop that holds the stored logic value, whereas the remaining pass transistors controlled by the wordline (WL) signal allow read and write operations to the cell through the bitline (BL) and its complement (BLB).

In a 6T cell, when the cell is stable and storing a "0," PU_R suffers from NBTI, and PD_L suffers from GTDDB. On the contrary, when the cell stores a "1," PU_L and PD_R are affected by NBTI and GTDDB, respectively. On the other hand, HCI affects all the transistors on a write if the logic value flips. Note that the wearout effects induced by each possible duty cycle are complementary, meaning that, for a given duty cycle, the pair of transistors not under stress are partially under recovery from NBTI degradation. Overall, the four transistors of the inverter loop are continuously aging regardless of whether the cell stores "0" or "1." This fact makes these transistors particularly sensitive to wearout [44]. Note that the pass transistors (nFETs) just age from BTI when the SRAM cell is being accessed, making them much less aging-sensitive than the inverter-loop transistors. Therefore, this article focuses on the wearout of the inverter-loop transistors.

C. Prior Art

SRAMs are highly sensitive to BTI-induced transistor-strength mismatch [45]–[47], and the FinFET SRAM is more vulnerable to BTI than the planar CMOS SRAM [36]. SRAM stability is analyzed in [48] and [49] by assuming two ideal stress conditions, static stress and alternating stress, and for a continuous range of stress states [50]. The SRAM degradation due to BTI based on a customer usage workload has been considered for logic [1], [51] and SRAM cells [2], [3], [52]–[54].

The shift of SRAM stability due to the HCI is less studied in prior research because BTI is usually dominant due to its frequency independence. However, since nowadays chips are running at higher frequencies, HCI is becoming an issue [55], [56]. In [57] and [58], the impact of HCI on SRAM cell stability is analyzed, and the simulation and experimental results were compared [57].

Methods that take into account usage scenarios have been proposed to mitigate aging. These methods include power gating [2], [3], [59], adaptive body bias [60], register address allocation [2], [53], and other techniques to more evenly distribute the stress [54]. Improved lifetime is achieved by balancing the amount of time that logic "0" and "1" values are stored in the cells with the aim to provide a BTI-optimal duty-cycle distribution [2], [54], [61], [62], and by implementing redundancy into the cache design to combat BTI-induced wearout [63]. It has also been proposed to mitigate the HCI degradation by providing a uniform distribution of cache accesses across sets of cells [54], [63].

Progressive gate–oxide BD is an important source of stability degradation in an SRAM. This is not only because of the thinner dielectrics, but also due to the lower supply voltage. In fact, it was found that leakage currents of 20–50 μA at the nFET source results in a 50% reduction in the noise margin (for 0.15 and 0.13 μm technologies) [64]. Moreover, an equivalent 100 K SBD resistance leads to an increase of 21% and 33% in delay and power of an SRAM cell, respectively [65].

RTN induces the erratic performance phenomenon and a higher failure probability [66]–[70]. With the scaling of technology, RTN adversely affects SRAM design margins [67]. Moreover, RTN reduces the read SNM and write margin by 12% and 3.9%, respectively [68]. In [69], RTN causes a 50% increase in the failure probability.

We study the impact of NBTI, HCI, GTDDB, and RTN on SRAM stability by checking the shifts of representative performance metrics. Based on the shift of performance metrics, we obtain the failure probability as a function of performance specifications. This article differs from prior work as we consider all wearout mechanisms simultaneously, together with realistic workloads. Our purpose is not to propose a wearout mitigation technique, but rather to demonstrate a methodology to evaluate the impact of wearout on caches in a realistic way so that designers can make appropriate tradeoffs.

III. Aging Assessment Framework

As the time-to-failure due to wearout is a function of device stress and the thermal profile, a framework for the acquisition of spatial thermal/electrical stress of a system was constructed. The FPGA-based emulation system extracts the duty-cycle/toggle-rate profiles and the temperature profile.

Running register transfer level (RTL) or Simulation Program with Integrated Circuit Emphasis (SPICE) simulations of a complete microprocessor to extract the activity profile of each SRAM cell is not feasible in most cases, since it may take a few months to simulate a single benchmark. On the other hand, simulating microprocessors with standard benchmarks on an FPGA takes only a few minutes. Our aging assessment framework provides an efficient way to acquire electrical and thermal profiles for any digital system for use in system-level reliability analysis. Any other emulation system, such as Gem5 [71], would be able to generate similar results after revising the source code carefully.

For analyzing the impact of BTI, HCI, GTDDB, and RTN on caches within a microprocessor system, we have implemented the well-known open-source LEON3 IP core processor [72] with superscalar abilities on various processes. The emulation system is implemented with an FPGA. Specifically, for activity tracking, the hardware RTL of the design was synthesized for the FPGA, and counters were placed at the I/O ports of the data cache, which track both state probabilities and toggle rates of the ports during application runtime. We used the LEON3 processor synthesized with the 90-nm process to extract the cache activity and then used the activity to study the aging of the cache implemented with a 14-nm FinFET process. The technology of the FPGA is not important. It is just

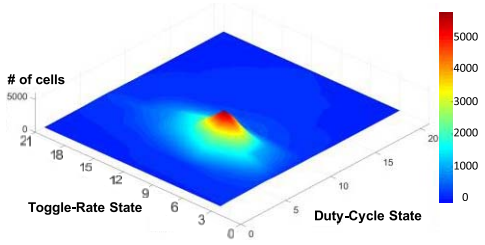

Fig. 2. Example of a stress-state distribution for a 32-kb memory. A stress state is a combination of the duty-cycle state and the toggle-rate state. The z-axis is the number of cells.

required that the FPGA has sufficient resources to implement the circuit being analyzed and that the FPGA implementation is functionally equivalent.

A set of standard benchmarks [73] were used as the applications for analysis. The outputs of the FPGA emulation are the duty cycle and toggle rate of the I/O's of the blocks of which the design is composed.

The microprocessor logic units consist of a 32-bit general-purpose integer unit (IU), a 32-bit multiplier (MUL), a 32-bit divider (DIV), and a memory management unit (MMU). Storage blocks, which are composed for SRAM cells, include a window-based register file unit (RF), separate data (D-Cache), and instruction (I-Cache) caches, and cache tag storage units (Dtags and Itags). In this article, we have focused on the L1 data cache due to its high activity and temperature. However, other cache blocks can be studied using the same methodology.

After emulation is complete, the I/O duty cycle, I/O toggle rate, and the netlist were then used for activity propagation to each SRAM cell in the data cache for a complete stress/transition probability profile of the SRAM arrays within microprocessor. This step takes into account the technology and netlist of the circuit being implemented. After activity propagation, we have the distributions of duty cycle and toggle rate for all of the SRAM arrays in the microprocessor. Fig. 2 shows an example joint distribution of the duty cycle and the transition rate of the data cache, when the microprocessor is running the Basicmath benchmark. Fig. 3 compares the duty cycle and transition rate for different memory blocks while running the same benchmark. BTI is sensitive to the duty-cycle distribution, and Fig. 3(a) shows that the data cache is the most vulnerable unit to BTI and the register file is the least vulnerable unit to BTI. HCI is sensitive to the toggle rate, and Fig. 3(b) shows that the register file is most sensitive to HCI and the data cache is least sensitive to HCI. Moreover, for a modern cache operating at a higher frequency, the most vulnerable block is more likely to be the register file.

Besides activity variation, temperature variation throughout the microprocessor is also taken into account when modeling the wearout mechanisms. The netlist was used for layout generation in the target technology (14 nm). The RC information from the layout, together with the net activities, was used for the extraction of the power profile and the consequent thermal profile, through the power (Synopsys PrimeTime) and the thermal (COMSOL) simulators, respectively, for every single block of the microprocessor. The COMSOL [74] heat transfer module determines the thermal distribution. The equivalent thermal resistance (R_{TH}) of 1.472e5 K/W due to FinFET

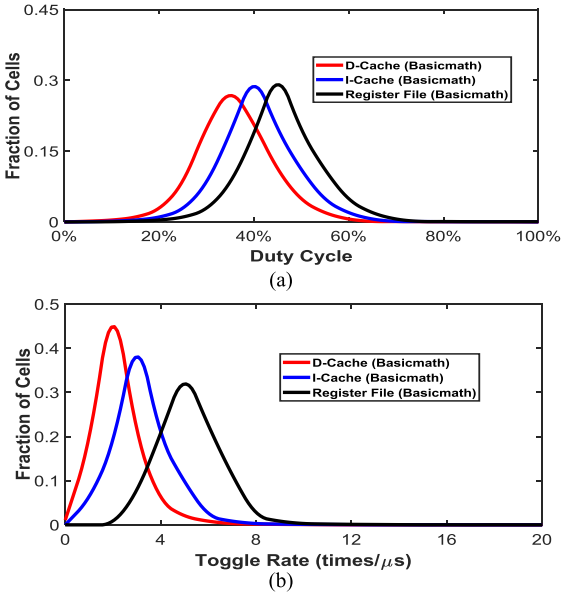

Fig. 3. Distribution of (a) duty cycle and (b) toggle rate for three memory blocks in the LEON3 microprocessor.

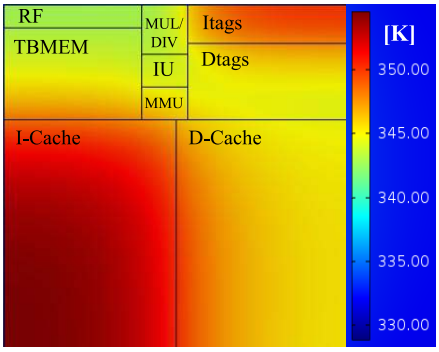

Fig. 4. Average temperature distribution of the microprocessor implemented in 14-nm technology while running the Basicmath benchmark.

self-heating (SH) is also incorporated [75]. Fig. 4 shows the average temperature distribution when the microprocessor implemented with 14-nm technology is running the Basicmath benchmark. During the microprocessor's operations, the activity and temperature are not constant. We have simulated the steady-state results while using the maximum power distribution to obtain a worst case estimate.

All the crucial blocks of a microprocessor are considered in this experiment. The power ratio for each block is calculated from PrimeTime. The I-Cache has the highest temperature in our simulations, because the operations of the microprocessor are determined by the instructions. Every time, when an instruction is executed, the I-Cache is involved, while the D-Cache is relatively less frequently involved. On the other hand, although Fig. 4 shows that the RF has a higher toggle rate than the I-Cache, the I-Cache has a higher temperature because of its much larger size and the larger number of bits involved during each operation. Meanwhile, Dtags has a lower temperature, since only a successful comparison of tags is accompanied with a read operation on the cache block, which involves more bits than the tags. Also, Dtags has a much smaller size than the D-Cache block.

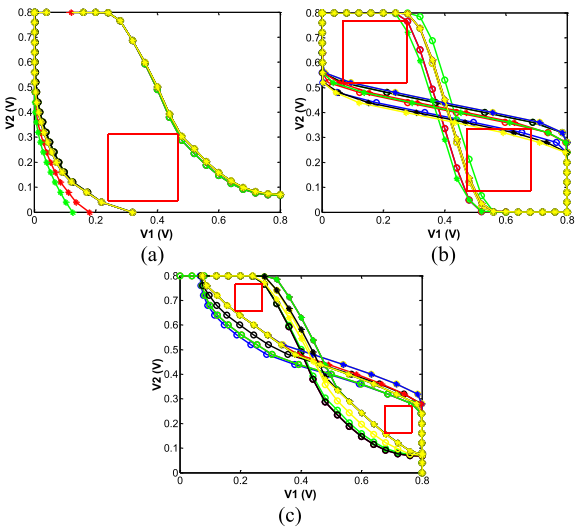

Fig. 5. DC sweep for (a) write, (b) hold, and (c) read SNMs, under various stress conditions. V1 and V2 represent the input and output voltages of the two inverters forming the inverter loop in the SRAM cell. All the simulations are obtained for 14-nm technology. The simulations vary depending on the gate length, duty cycle, transition rate, and temperature randomly. Ten sample curves are shown. The results confirm that the read SNM is the smallest (most important) for all possible sets of parameters.

IV. SRAM Reliability Characterization

A. Performance Degradation Analysis

SRAMs are characterized with several performance metrics. These include the read/write/hold SNMs, minimum voltage for state retention (vdd-min-ret), read delay and power, write delay and power, and the leakage power during hold.

The SNMs are defined as the minimum dc noise voltage necessary to change the state of an SRAM cell.

The stability margins are extracted by fitting squares between the SNM curves and observing the diagonal length of the smaller of the two squares [76]. While measuring the SNM, the inverter loop is first opened to form two inverters, and then the input–output curves are obtained by sweeping the input voltage. The SNM is calculated from fitting the squares between the input–output curves. The read margin is measured with the access transistors turned ON and BL/BLB at Vdd (the supply voltage). The write margin is the minimum voltage needed to flip the state of the cell, with the access transistors turned on and BL/BLB at their own voltages (0/1 for write 0, and 1/0 for write 1). The hold margin is measured when the access transistors are turned off. Vdd-min-ret is the minimum voltage in which the SRAM retains its state. Finally, delay and power are also extracted with SPICE simulations.

Fig. 5 shows the input–output curves of the SRAM inverters, with squares inserted to measure the noise margins (write, hold, and read). As shown in Fig. 5, the read SNM is the smallest of the SNMs for various device dimensions (gate length) and stress conditions (duty cycle, transition rate, temperature). Therefore, we only consider the read SNM among the SNMs when determining the cell failure probability in the next step.

With the increase in degradation, when any of the seven metrics degrades to a predefined threshold, the SRAM cell

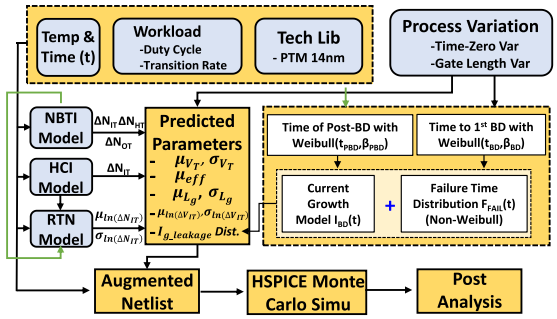

Fig. 6. Framework for MC simulation for FinFET SRAM cell degradation.

is considered to have failed. In this article, MC simulations in HSPICE were implemented to obtain 2000 samples for each performance metric for each stress condition considered. Fig. 6 shows our detailed simulation flow. First, the cell's device information, workload, temperature, and stress time are fed to front-end wearout mechanism models to obtain the corresponding parameters, such as trap density shifts and distributions and the T_{BD} relevant to GTDDB. Then the failure probability based on time to the first BD distribution and the time to post-BD distribution are calculated with (8). The exact leakage current distribution is obtained from the current growth model and the GTDDB failure probability. The ratio of leakage current from the gate to the drain versus the gate to the source is assumed to be uniformly distributed between 0 and 1 for each device [35]. Third, the SRAM cell netlist is augmented with parameters predicted by the degradation mechanisms. Finally, the HSPICE MC simulations are performed to get various performance degradation distributions. Post analysis is needed for data display and to compute the cells' failure probability.

Fig. 7 shows the degradation of the read SNM, Vdd-min-ret, read delay and power, write delay and power, and hold (leakage) power of a memory cell with a 20% duty cycle and a 10 transitions/μs transition rate. It shows a comparison between degradation due to all front-end mechanisms, due to NBTI and GTDDB, and due to NBTI and HCI. At the 10^6 s time point, NBTI and HCI severely degrade the read SNM and Vdd-min-ret and improve hold power, while read delay and power and write delay and power are relatively unaffected. When NBTI and GTDDB are considered simultaneously, both read SNM and Vdd-min-ret degrade, while the other performance metrics are not obviously affected. The different shift direction of Vdd-min-ret introduced by GTDDB is caused by the fact that the inverters' state is easier to flip if the pull-down (PD) nFETs have a higher gate leakage current. Therefore, the overall Vdd-min-ret gets higher (worse) with GTDDB.

When NBTI, HCI, and GTDDB are present, the read SNM and Vdd-min-ret degrade more in comparison with NBTI and GTDDB. We should also note that the HCI effect has a strong dependence on the operating frequency. Throughout this article, the LEON3 microprocessor is set to run at 250 MHz [76]. For this situation, BTI is dominant and HCI has a smaller influence. HCI would have more influence when the operating frequency reaches the GHz range.

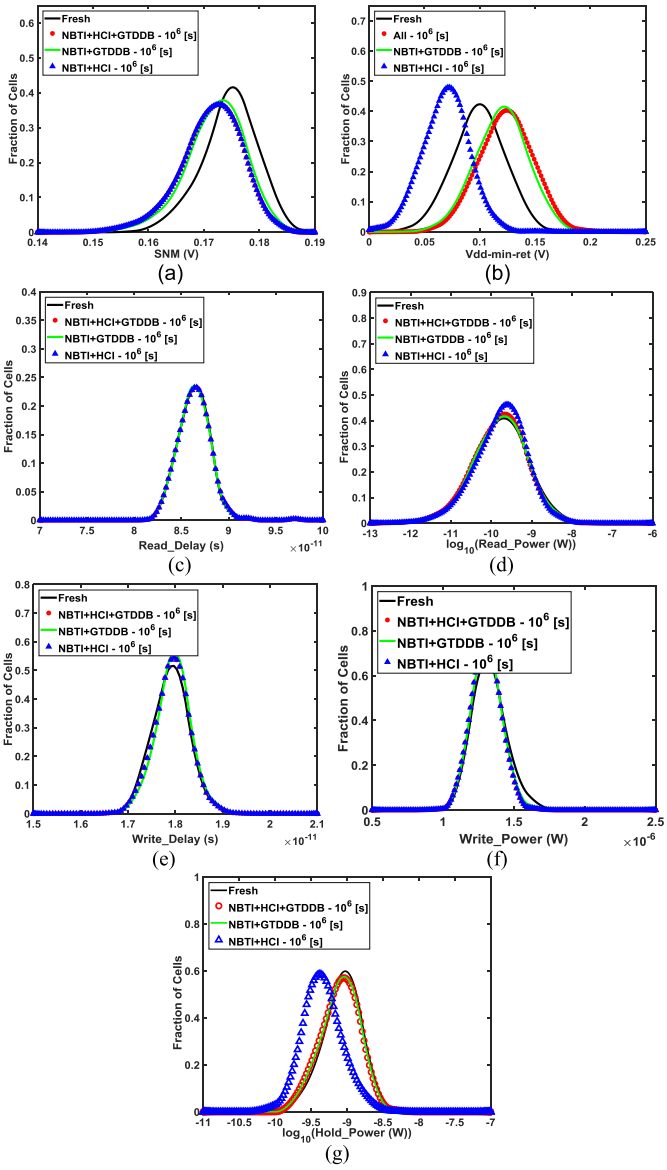

Fig. 7. Degradation of (a) read SNM, (b) Vdd-min-ret, (c) read delay, (d) read power, (e) write delay, (f) write power, and (g) hold power of SRAM cells under different combinations of wearout mechanisms. All curves assume that the cell has experienced a 20% duty cycle and a 10 transitions/μs transition rate. All simulations were obtained for 14-nm technology.

B. Memory PF Characterization

When any of the performance metrics mentioned in Section IV-A degrades to a predefined threshold, the SRAM cell is said to have failed, and thus the PF of the cell is obtained. Using MC simulations, the PF of an SRAM cell at each time point is obtained for the given performance distributions and the constraints.

Running SPICE simulations for each SRAM cell is very computationally expensive. To manage the large volume of SRAM cells and to limit the number of SPICE simulations, we partition both static stress probability and switching activity into states to balance the accuracy and computational cost of the simulations for NBTI, GTDDB, HCI, and RTN. The cells in the same stress state have the same degradation.

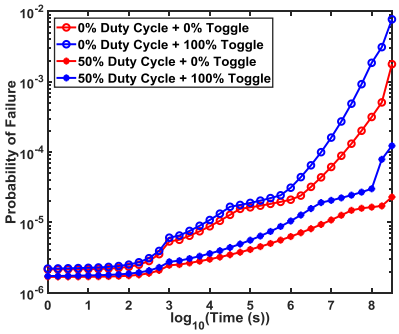

Fig. 8. Failure probability evolution of an SRAM cell when it is in a specific stress state. Each stress state is a combination of the duty-cycle state and the toggle-rate state. Four stress states are shown in this figure. Note that the figure shows limiting cases, which do not occur in practice (see Fig. 3). For example, "0% duty cycle + 100% toggle" and "50% duty cycle and 0% toggle" are such limiting conditions. These cases are included to illustrate the trend toward these extremes.

For NBTI and GTDDB, the stress states represent duty cycles. 0% duty cycle means the cell has 0% time storing a "1," while 100% duty cycle corresponds to 100% time storing a "1." For HCI, the stress states are proportional to the maximum observed transition rate, a fraction of the maximum transition rate. One example of such a distribution of stress states is illustrated in Fig. 2, for a 32-kb data cache. Note that the stress distribution not only depends on the applications being run, but also depends on the memory allocation of the cache system, which will be discussed in detail in Section V. When NBTI, GTDDB, and HCI are combined, the stress states are combinations of the duty-cycle state and the toggle-rate state. For example, a stress state could have a low duty cycle and a high toggle rate, or a high duty cycle and a low toggle rate.

Since degradation and process variations are considered, the performance degradation of each SRAM cell is a distribution rather than a fixed value. Because we consider all the cells in the same stress state to have the same stress, all the cells in one stress state share the same degradation distribution at each stress time point. By running MC simulations in HSPICE, the performance distribution is computed for each stress state at each time point. The PF of each cell is calculated as the fraction of performance distributions whose values exceed their predefined threshold values. An example of a failure probability after various stress times for four different stress states is illustrated in Fig. 8 for the combined effect of NBTI, HCI, GTDDB, and RTN.

As can be seen from this figure, an SRAM cell has a lower probability of failure when it has a 50% duty cycle and a 0% toggle rate. Moreover, higher toggle rates result in a worse failure probability. The failure probability distribution of all stress states is characterized for two different temperatures, that is, temperature is partitioned into two states.

The PF of a word is then calculated by

$$\mathrm{PF}_{\mathrm{word}} = 1 - \prod_{i=1}^{N} (1 - \mathrm{PF}_{\mathrm{bit}i}) \tag{14}$$

where $\mathrm{PF}_{\mathrm{word}}$ is the PF of a word, $\mathrm{PF}_{\mathrm{bit}}$ is the PF of a bit, and N is the number of bits in one word. The word size is $N = 32$

for the data cache of the LEON3. Since PF_{bit} changes as a function of time, PF_{word} also changes as a function of time.

If the SRAM does not use error-correcting codes, the memory fails when the first cell fails to work. The PF of the SRAM block is obtained accordingly as a function of time

$$PF_{SRAM} = 1 - \prod_{i=1}^{N_{word}} (1 - PF_{bit\,i}) \tag{15}$$

where PF_{SRAM} is the PF of the whole memory block, PF_{word_i} is the PF of word i and N_{word} is the number of words.

Error-correcting codes (ECCs) can ensure that a memory system can tolerate faults. Bose-Chaudhun-Hocquenghem (BCH) codes [77] require seven additional bits per word and can correct one bit per word. The relationship between failures of single bits, P_{fail}, and the failure of the word is modeled with a binomial distribution. For a word containing N bits, the PF of a word, F_{word}, is

$$PF_{word} = 1 - \prod_{i=1}^{N}(1 - PF_{bit\,i}) - \sum_{j=1}^{N}\left[PF_{bit\,j} * \prod_{i\neq j}(1 - PF_{bit\,i})\right]. \tag{16}$$

The word size when there are ECCs is $N = 39$ for the D-Cache, I-Cache, and RF blocks of the LEON3. The failure probability of the memory, PF_{SRAM}, is calculated using (15).

V. Performance–Reliability Analysis For Different Cache Configurations

Based on the method for memory lifetime characterization in Section IV, the reliability (failure probability) of the LEON3 L1 data cache was studied for different cache configurations: associativity, cache line size, and cache size. Since the least-recently-used (LRU) replacement algorithm is found to cause the highest failure probability compared with the least recently replaced (LRR) and Random algorithms, we take it as the default setting [78]. The impact of ECCs is also analyzed.

Six representative benchmarks from MiBench [73] were run on the microprocessor: Basicmath, Qsort, SHA, cyclic redundancy check (CRC) 32, FFT, and Dijkstra. These may not be the most advanced benchmarks, but they easily run on the LEON3 microprocessor and are sufficient to illustrate our methodology. The Basicmath benchmark performs simple mathematical calculations that often do not have dedicated hardware support in embedded processors. Qsort sorts a large array of strings into ascending order using the well-known quick sort algorithm. SHA is the secure hash algorithm that produces a 160-bit digest for a given input. CRC32 is a benchmark performing a 32-bit CRC on a file to detect errors in data transmission. FFT performs a Fast Fourier Transform (FFT) on an array of data. The Dijkstra benchmark constructs a large graph in an adjacency matrix representation and then calculates the shortest path between every pair of nodes using the repeated application of Dijkstra's algorithm.

The aging assessment framework in Section III was used to extract the duty-cycle/toggle-rate and temperature of the data cache for different cache designs and for the six applications above. The method for memory lifetime characterization described in Section IV was then used to calculate the failure probability of the data cache while running each application.

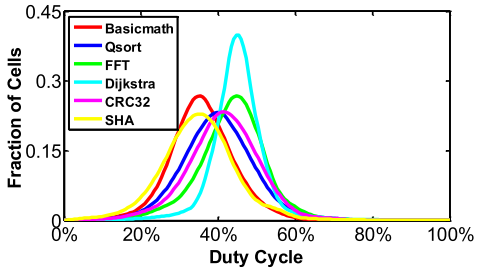

Fig. 9. Duty-cycle distributions of SRAM cells in a two-way 32-kb data cache while the microprocessor is running six different benchmarks.

Fig. 9 shows the duty-cycle distribution for each application using a two-way 32-kb data cache with a 16-byte line size. Clearly, logic "0" is the predominant state.

Memories in a processor contain more 0s than 1s throughout normal operations [79]. In general, "0" is stored longer than "1" because the memory is usually initialized to zero when it is allocated. Thus, even if there is an equal likelihood of an application writing a "0" or a "1" in any bit position, this initialization always means that "0" is stored longer. Other reasons for "0" being stored longer are that false Boolean values and NULL pointers are represented with zero, as well as most data in dense-form sparse matrices [77].

According to the performance requirements and the reliability budget, cache designers could optimize a cache by balancing performance, reliability, and area requirements. We define a performance metric HPS which is a function of HR, PF, and cache area (Area)

$$HPS = \frac{(A_{HR}HR - B_{HR})^{n_{HR}}}{(A_{PF}PF - B_{PF})^{n_{PF}}(A_{Area}Area - B_{Area})^{n_{Area}}}. \tag{17}$$

In general, our target is to obtain a high HR and low PF at a suitable cache area. We set the constants according to the importance of the various requirements, so that a higher HPS results in a better design. Therefore, a cache designer can optimize HPS to achieve the best possible design. In (17), A_{HR}, B_{HR}, n_{HR}, A_{PF}, B_{PF}, n_{PF}, A_{Area}, B_{Area}, and n_{Area} are constants that can be adjusted based on the design requirements. In our study, A_{HR}, n_{HR}, A_{PF}, B_{PF}, n_{PF}, B_{Area}, and n_{Area} are fixed at $1.0e^{-4}$, 5.0, $1.0e^{2}$, 0, 0.1, 0, and 1.5, respectively. B_{HR} and A_{Area} are chosen to be different for each benchmark to make the impact of configuration parameters under various benchmarks observable in a similar range. They are selected as 9.6e3 and 0.7629 μm^{-2} for Basicmath, 7.58e3 and 1.63e2 μm^{-2} for CRC32, 6.2e3 and 8.54e2 μm^{-2} for Dijkstra, 7.24e3 and 4.11e2 μm^{-2} for FFT, 7.8e3 and 1.16e2 μm^{-2} for Qsort, and 9.74e3 and 0.2154 μm^{-2} for SHA, respectively.

A. Associativity

Cache associativity can be taken as the method to select bookshelves of different shapes and sizes. Caches fall into one of three categories: direct mapped, n-way set associative, and fully associative. Direct mapped caches are designed so that

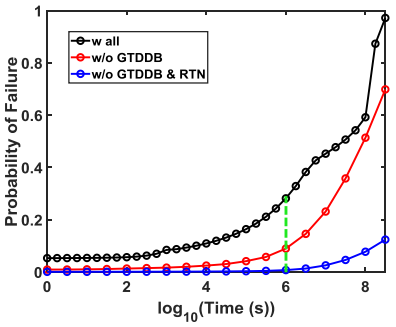

Fig. 10. Failure probability as a function of time for one-way associativity with the Basicmath benchmark while all the wearout mechanisms are included, GTDDB is excluded, and both GTDDB and RTN are excluded.

a cache block can only go in one spot in the cache. Two-way set associative caches are made up of sets such that each set can fit into two blocks, while in a four-way set associative cache, each set fits into four blocks. For a fully associative cache, a cache block can go anywhere in the cache. It is worth noting that the directly mapped cache is actually a one-way set associative cache and a fully associative cache of m blocks is an m-way set associative cache. Higher associativity improves the HR, but reduces the cycle time and costs more area because of the need for more comparators. The L1 data cache of the LEON3 was implemented with three different associativities: one-way, two-way, and four-way, while the cache line size (16 byte), cache size (32 kb), and the replacement algorithm (LRU) were kept the same.

Fig. 10 shows the PF as a function of time for one-way associativity under the Basicmath benchmark, with different combinations of wearout mechanisms. According to a comparison of the PF, it is found that GTDDB and RTN affect reliability significantly. GTDDB impacts SRAM reliability by introducing leakage currents which impact device performance metrics and cause them to shift more easily. As a result, the SRAM becomes more sensitive to NBTI and HCI. Similarly, since the interface trap density due to NBTI and HCI accumulates, the impact of RTN on the failure probability increases with stress time. Therefore, it is necessary to include all the wearout mechanisms in simulations.

Fig. 11 shows the comparison of failure probability at 10^6 s for three associativities and two benchmarks. For illustration purposes, the results from two applications are shown: Basicmath and Dijkstra, since other applications produce the same trend. Note that the impact of associativity on the failure probability is highly related to the mechanisms considered, while the impact of associativity is small.

The HRs for one-way, two-way, and four-way associativities are 96.12%, 96.33%, 96.36%, respectively, for Basicmath, and are 62.23%, 64.81%, 65.54%, respectively, for Dijkstra. Higher associativity results in a higher HR, but also increases the failure probability. A higher HR produces fewer misses, and thus the cells are more likely to keep their stored values unchanged, which aggravates NBTI and GTDDB. From the perspective of aging, a cache miss is potentially useful as it flips the value stored in a cell and mitigates NBTI and GTDDB.

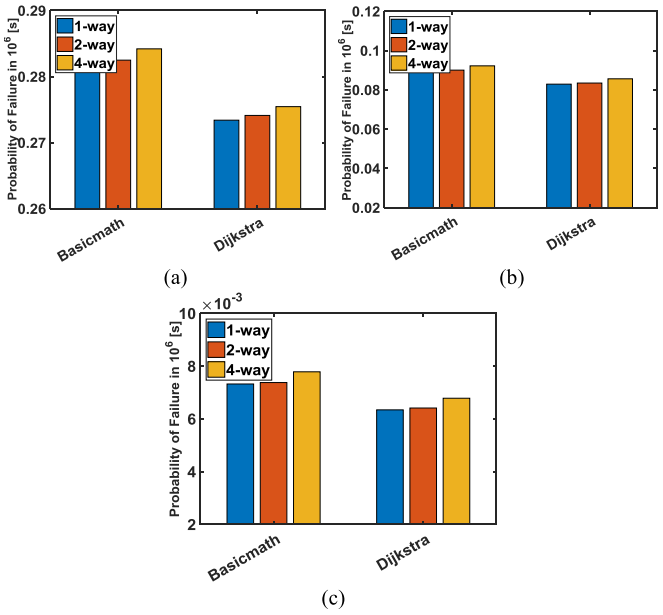

Fig. 11. Comparison of the failure probability at 10^6 s for the three associativity algorithms and two benchmarks with (a) all wearout mechanisms included, (b) GTDDB excluded, and (c) both GTDDB and RTN excluded.

HPS varies with associativity under different benchmarks. For example, when the benchmark is Basicmath with one-way associativity, the values of HPS are 0.123, 0.138, and 0.177. When the associativity is two-way, the values of HPS are 19.35, 21.7, and 27.86, respectively. When the associativity is four-way, the values of HPS are 29.88, 33.44, and 42.82, respectively. Four-way associativity is the optimal option with respect to HPS. The trend is similar for the Dijkstra benchmark.

B. Cache Line Size

Data is transferred between the main memory and the cache in blocks of fixed size, called cache lines. When a cache line is copied from the main memory into the cache, a cache entry is created. The cache entry includes the copied data as well as the requested memory location (called a Tag). We implemented the data cache with two different cache line sizes: 16- and 32 byte, while the two-way associativity, cache size (32 kb), and the replacement algorithm (LRU) are kept the same.

Fig. 12 shows the percentage reduction of the PF at 10^6 s and the percentage improvement of the HR of the six benchmarks with a 32-byte cache line compared with a 16-byte cache line. The simulation results with all mechanisms considered, with GTDDB excluded, and with GTDDB & RTN excluded, are shown. Obviously, it is necessary to include all wearout mechanisms in the analysis. It is observed that the 32-byte cache line has a lower failure probability than the 16-byte cache line for all six benchmarks. When including all wearout mechanisms, the 32-byte cache line always has an improved failure probability in comparison with the 16-byte cache line. Except for Basicmath and SHA, where there is almost no hit-rate improvement, the 32-byte cache line also achieves a better performance than the 16-byte cache line.

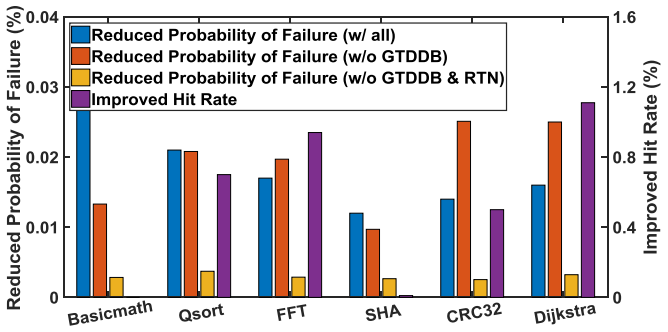

Fig. 12. Percentage reduction in the PF at 10^6 s and the percentage improvement in the HR, for six applications, with all wearout mechanisms included, GTDDB excluded, and both GTDDB and RTN excluded. The improved/reduced value is defined as the improvement/reduction of using a 32-byte cache line compared to a 16-byte line.

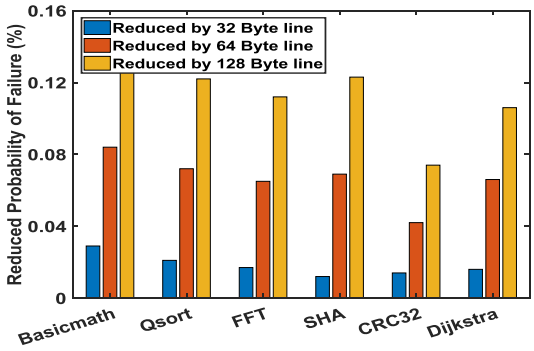

Fig. 13. Percentage reduction of the PF at 10^6 s, for six applications, with all wearout mechanisms included. The percentage reduction is defined for cache line sizes with the 32-, 64-, and 128-byte cache lines in comparison with the 16-byte line.

It can be seen that compared to the case without GTDDB, the PF reduction is lower for the case with all the wearout mechanisms for some benchmarks. The reason is that although a larger cache line size causes the overall duty cycle distributions to get closer to 50%, the sensitivity of the PF reduction for cells with a duty cycle close to 50% is different for the cases with all wearout mechanisms included and the cases with GTDDB excluded. It's found that when the duty cycle is very close to 50%, the PF reduction of cells with all wearout mechanisms considered is lower than that of the cases with GTDDB excluded.

Overall, the 32-byte cache line is better than the 16-byte cache line in both performance and reliability, although this improvement is not very large. This observation is a little counter-intuitive as we have shown that a higher HR results in lower reliability in Section V-A. So, it might be straightforward to think that the 32-byte cache line would have a higher failure probability because of its higher HR. However, a cache miss in a 32-byte cache line produces recovery cycles for up to 256 (32×8) SRAM cells, which is twice as many as with a 16-byte cache line (16×8 SRAM cells). Therefore, although a 32-byte cache line has fewer misses, it actually has a larger number of NBTI stress recovery cycles than a 16-byte cache line, which results in improved reliability. Fig. 13 shows the percentage reduction of the PF at 10^6 s, with 32-, 64-, and 128-byte cache lines when compared with a 16-byte cache line. The larger cache line size helps improve the PF.

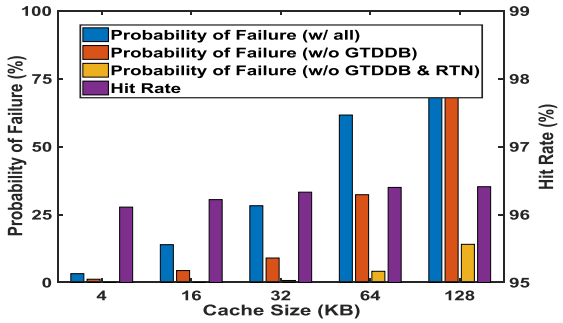

Fig. 14. Hit rate and the PF at 10^6 s for five different cache sizes under the Basicmath application, with all wearout mechanisms included, GTDDB excluded, and both GTDDB and RTN excluded.

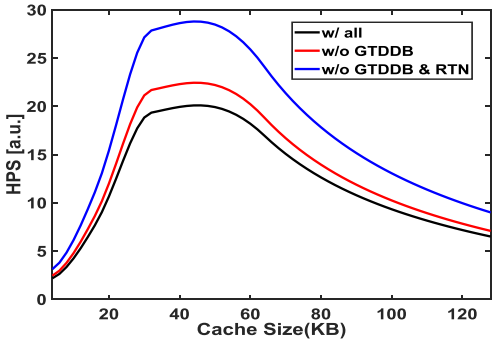

Fig. 15. HPS as a function of cache size, with all wearout mechanisms included, GTDDB excluded, and both GTDDB and RTN excluded. The benchmark is Basicmath.

The differences among applications for cache line size are because the activity shifts differently under various applications. Since different applications cause different data to be kept in the cache, the impact on the cells' activity is not always the same. The improved PF also varies with application.

Obviously, cache line size impacts HPS. If the benchmark is FFT, the values of HPS are 19.94, 118.93, 139.21, and 188.02 with line sizes of 16, 32, 64, and 128 bytes, respectively. The trend is similar for the benchmark Qsort.

C. Cache Size

The size of the data cache is another important metric in cache system design. In our experiments, the data cache was implemented with five different cache sizes: 4, 16, 32, 64, and 128 kb, while the associativity (two-way), cache line size (16 byte), and replacement algorithm (LRU) were kept the same. Fig. 14 shows the HR and failure probability (at 10^6 s) for five different cache sizes and the Basicmath application. The failure probability increases dramatically as the cache size increases because the failure probability is larger when there are more SRAM cells.

It is also observed that the HR increases as the cache size increases. However, when the cache size is larger than 32 kb, little improvement is seen in the HR.

According to the performance requirements and the reliability budget, cache designers could determine an optimal cache size balancing both performance and reliability requirements, by maximizing HPS, as shown in Fig. 15. As can be seen from the figure, we may not prefer a very large cache because large

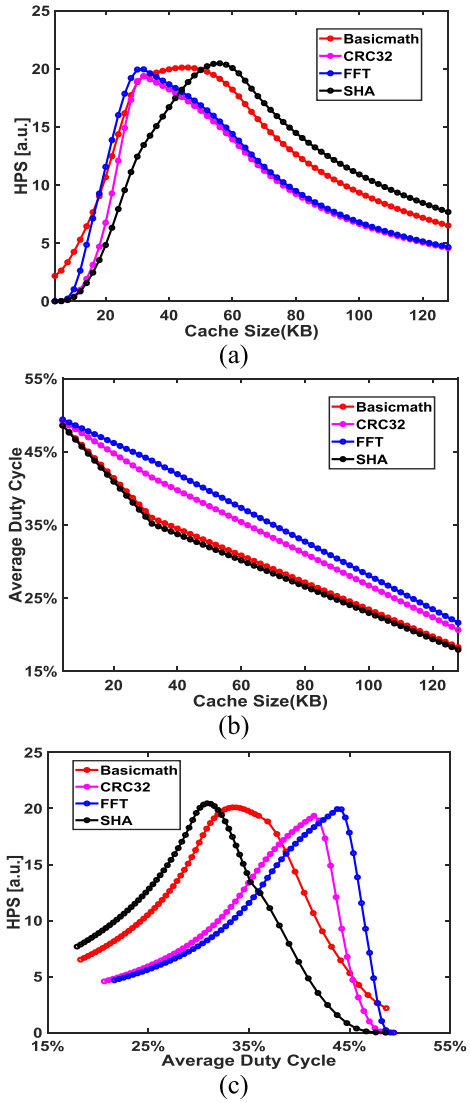

Fig. 16. (a) HPS as a function of cache size, (b) average duty cycle as a function cache size, and (c) HPS as a function of average duty cycle, for four applications, with all wearout mechanisms included.

caches result in a large area and high power consumption. According to the HPS distribution in Fig. 15, we can say that a 32-kb cache size provides an optimal design. When including different wearout mechanisms, the optimal solution does not change. The parameters of the HPS function are just an example, and the methodology can be extended to study more complicated cases.

Fig. 16(a)–(c) shows HPS as a function of cache size, the average duty cycle as a function of cache size, and HPS as a function of the average duty cycle for four applications, respectively, while the associativity (two-way), cache line size (16 byte), and replacement algorithm (LRU) were kept the same. In Fig. 16(c), HPS is a function of the average duty cycle, which is obtained from Fig. 16(a) and (b), by sweeping cache size from small to large. Here, all the wearout mechanisms are included. First, it is found that the cache size for optimal HPS is impacted by the application. Second, the average duty cycle always gets lower with the increase in cache size. Third, the cache size for optimal HPS gets higher

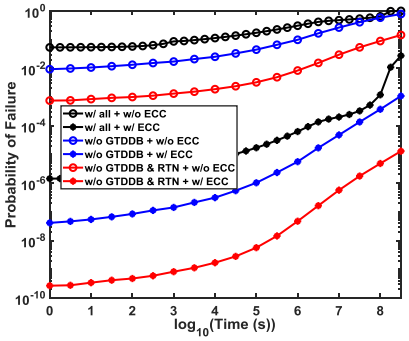

Fig. 17. PF of the two-way 32-kb data cache with and without ECCs as a function of time for the Basicmath application, with all wearout mechanisms included, GTDDB excluded, and both GTDDB and RTN excluded.

if the overall duty-cycle distribution gets lower. The optimal HPS is affected by the average duty cycle of the applications.

D. Error-Correcting Codes

ECCs detect and correct the most common sources of internal data corruption. ECCs add some redundancy (some extra bits) to check the consistency of the data and to recover corrupted data. The word size containing the ECC codes for the data cache of the LEON3 is N = 39 when the implemented ECCs are designed to correct single-bit errors.

The failure probabilities of the two-way 32-kb data cache with and without ECCs are shown in Fig. 17 as a function of time, with all the wearout mechanisms included, GTDDB excluded, and both GTDDB and RTN excluded. The Basicmath benchmark is illustrated as an example, and other benchmarks produce similar results. It can be seen that ECCs lead to a substantial improvement in the failure probability. In fact, increasing the number of bits corrected always lowers the PF.

Although ECCs lower the PF, the area increases. For the example shown in Fig. 18, when ECCs are not considered, the values of HPS are 19.21, 21.51, and 27.53 with all wearout mechanisms included. With ECCs, the values of HPS are 33.38, 42.5, and 68.5 with all wearout mechanisms included. On the basis of the case considered here, ECCs improve HPS.

VI. Conclusion

NBTI, HCI, GTDDB, and RTN progressively degrade the parameters of transistors, such as threshold voltage, carrier mobility, fringe capacitance, gate leakage current, resulting in stability degradation of SRAMs. These wearout mechanisms are especially critical for the transistors of first-level (L1) caches, which are frequently accessed and continuously aging.

In this article, we have studied the impact of a variety of wearout mechanisms on SRAM lifetime. The wearout mechanisms are sensitive to two parameters which are a function of the application running on the microprocessor, the duty cycle and the toggle rate. We have found that if the application running on the SRAM has a duty-cycle distribution closer to 50% and a toggle rate closer to 0%, the PF is lower. Hence, applications whose characteristics are closer to these limits have better lifetimes.

We have also presented the reliability and performance of the data cache while varying four different configuration parameters: associativity, cache line size, cache size, and ECCs. A general rule is that higher performance (higher HR) results in lower reliability (higher failure probability). This is because the additional resources needed to improve the HR degrade lifetime.

One exception of this rule happens for different cache line sizes. The 32-byte cache line is better than the 16-byte cache line in both performance and reliability. One cache miss for a 32-byte cache line can recover as many as 256 (32×8) cells from BTI stress, which is twice as many as for a 16-byte cache line. Therefore, despite the fact that the 32-byte cache line has fewer cache misses, it actually results in more NBTI stress recovery. Thus, the 32-byte cache line achieves higher performance and better reliability.

Cache size is of great significance to both cache performance and reliability. It is observed that when the cache size increases to larger than 16 kb, the cache reliability dramatically drops, while the performance (HR) shows very limited improvement. Moreover, ECCs improve reliability at the cost of area and power overhead. The two configuration parameters that most strongly impacted reliability are cache size and ECCs.

It is important to include all of the wearout mechanisms while evaluating the PF of a cache. Then, based on the user-defined importance of cache area, HR, and PF, we can obtain the target cache configuration which results in an optimal HPS.

Overall, the proposed framework can efficiently evaluate the performance and reliability of the cache memory and can provide insights to help cache designers optimize performance–reliability tradeoffs by selecting the appropriate cache configurations based on the specification budget and lifetime requirements. Instead of just determining whether the duty-cycle distribution is closer to 50%, users can define requirements on HR, PF, and cache size to obtain the best cache configuration at a specific stress time.

REFERENCES

[1] D. Kraak *et al.*, "Methodology for application-dependent degradation analysis of memory timing," in *Proc. DATE*, Mar. 2019, pp. 162–167.

[2] A. Valero, F. Candel, D. Suárez-Gracia, S. Petit, and J. Sahuquillo, "An aging-aware GPU register file design based on data redundancy," *IEEE Trans. Comput.*, vol. 68, no. 1, pp. 4–20, Jan. 2019.

[3] M. Namaki-Shoushtari, A. Rahimi, N. Dutt, P. Gupta, and R. K. Gupta, "ARGO: Aging-aware GPGPU register file allocation," in *Proc. CODES+ISSS*, Oct. 2013, p. 30.

[4] N. P. Jouppi, "Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers," in *Proc. ISCA*, May 1990, pp. 364–373.

[5] D. H. Albonesi, "Selective cache ways: On-demand cache resource allocation," in *Proc. MICRO*, Nov. 1999, pp. 248–259.

[6] A. Jalee, K. B. Theobald, S. C. Steely, Jr., and J. Emer, "High performance cache replacement using re-reference interval prediction (RRIP)," in *Proc. ISCA*, Jun. 2010, pp. 60–71.

[7] M. A. Alam, "A critical examination of the mechanics of dynamic NBTI for PMOSFETs," in *IEDM Tech. Dig.*, Dec. 2003, pp. 14.4.1–14.4.4.

[8] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "An analytical model for negative bias temperature instability," in *Proc. IEEE/ACM ICCAD*, Nov. 2006, pp. 493–496.

[9] T. Grasser, W. Gos, and B. Kaczer, "Dispersive transport and negative bias temperature instability: Boundary conditions, initial conditions, and transport models," *IEEE Trans. Device Mater. Rel.*, vol. 8, no. 1, pp. 79–97, Mar. 2008.

[10] B. Kaczer *et al.*, "Atomistic approach to variability of bias-temperature instability in circuit simulations," in *Proc. IEEE IRPS*, Apr. 2011, pp. XT.3.1–XT.3.5.

[11] H. Kükner *et al.*, "Comparison of reaction-diffusion and atomistic trap-based BTI models for logic gates," *IEEE Trans. Device Mater. Rel.*, vol. 14, no. 1, pp. 182–193, Mar. 2014.

[12] K. Joshi, S. Mukhopadhyay, N. Goel, and S. Mahapatra, "A consistent physical framework for N and P BTI in HKMG MOSFETs," in *Proc. IEEE IRPS*, Apr. 2012, pp. 5A.3.1–5A.3.10.

[13] N. Goel, T. Naphade, and S. Mahapatra, "Combined trap generation and transient trap occupancy model for time evolution of NBTI during DC multi-cycle and AC stress," in *Proc. IEEE IRPS*, Apr. 2015, pp. 4A.3.1–4A.3.7.

[14] N. Parihar, U. Sharma, R. G. Southwick, M. Wang, J. H. Stathis, and S. Mahapatra, "Ultrafast measurements and physical modeling of NBTI stress and recovery in RMG FinFETs under diverse DC-AC experimental conditions," *IEEE Trans. Electron Devices*, vol. 65, no. 1, pp. 23–30, Jan. 2018.

[15] S. Ramey *et al.*, "Intrinsic transistor reliability improvements from 22 nm tri-gate technology," in *Proc. IEEE IRPS*, Apr. 2013, pp. 4C.5.1–4C.5.5.

[16] K. T. Lee *et al.*, "Technology scaling on high-K metal-gate FinFET BTI reliability," in *Proc. IEEE IRPS*, Apr. 2013, pp. 2D.1.1–2D.1.4.

[17] N. Goel, P. Dubey, J. Kawa, and S. Mahapatra, "Impact of time-zero and NBTI variability on sub-20nm FinFET based SRAM at low voltages," in *Proc. IEEE IRPS*, Apr. 2015, pp. CA.5.1–CA.5.7.

[18] S. Ramey, J. Hicks, L. S. Liyanage, and S. Novak, "BTI recovery in 22nm tri-gate technology," in *Proc. IEEE IRPS*, Jun. 2014, pp. XT.2.1–XT.2.6.

[19] Y.-H. Lee, J. H. Lee, Y. S. Tsai, S. Mukhopadhyay, and Y. F. Wang, "Modeling of BTI-aging V_T stability for advanced planar and FinFET SRAM reliability," in *Proc. Int. Conf. Simulation Semicond. Processes Devices (SISPAD)*, Sep. 2017, pp. 85–88.

[20] A. Herrera-Moreno, J. L. García-Gervacio, H. Villacorta-Minaya, and H. Vázquez-Leal, "TCAD analysis and modeling for NBTI mechanism in FinFET transistors," *IEICE Electron. Express*, vol. 15, no. 14, pp. 1–12, 2018.

[21] H. Xie, X. Wu, Z. Peng, and H. Zhang, "The energy criterion for breaking chemical bonds in electrical breakdown process of polymers," in *Proc. ICPADM*, vol. 1, Jul. 1994, pp. 39–41.

[22] N. D. Akhavan, I. Ferain, R. Yu, P. Razavi, and J.-P. Colinge, "Emission and absorption of optical phonons in multigate silicon nanowire MOSFETs," *J. Comput. Electron.*, vol. 11, no. 3, pp. 249–265, Sep. 2012.

[23] A. N. Tallarico *et al.*, "Impact of the substrate orientation on CHC reliability in n-FinFETs—Separation of the various contributions," *IEEE Trans. Device Mater. Rel.*, vol. 14, no. 1, pp. 52–56, Mar. 2014.

[24] C. D. Young *et al.*, "Hot carrier degradation in HfSiON/TiN fin shaped field effect transistor with different substrate orientations," *J. Vac. Sci. Technol. B, Microelectron. Nanometer Struct. Process., Meas., Phenomena*, vol. 27, no. 1, pp. 468–471, 2009.

[25] I. Messaris *et al.*, "Hot carrier degradation modeling of short-channel n-FinFETs suitable for circuit simulators," *Microelectron. Reliab.*, vol. 56, pp. 10–16, Jan. 2016.

[26] Y. Wang, S. Cotofana, and L. Fang, "A unified aging model of NBTI and HCI degradation towards lifetime reliability management in nanoscale MOSFET circuits," in *Proc. Int. Symp. Nanosc. Archit.*, 2011, pp. 175–180.

[27] M. Jin *et al.*, "Hot carrier reliability characterization in consideration of self-heating in FinFET technology," in *Proc. IEEE IRPS*, Apr. 2016, pp. 2A.2.1–2A.2.4.

[28] E.-A. Chung *et al.*, "Investigation of hot carrier degradation in bulk FinFET," in *Proc. IEEE IRPS*, Apr. 2017, pp. XT-6.1–XT-6.4.

[29] S. Guo *et al.*, "Towards reliability-aware circuit design in nanoscale finFET technology: New-generation aging model and circuit reliability simulator," in *Proc. IEEE/ACM ICCAD*, Nov. 2017, pp. 780–785.

[30] S. Tous, E. Y. Wu, and J. Suñé, "A compact model for oxide breakdown failure distribution in ultrathin oxides showing progressive breakdown," *IEEE Electron Device Lett.*, vol. 29, no. 8, pp. 949–951, Aug. 2008.

[31] S. Mishra and S. Mahapatra, "On the impact of time-zero variability, variable NBTI, and stochastic TDDB on SRAM cells," *IEEE Trans. Electron Devices*, vol. 63, no. 7, pp. 2764–2770, Jul. 2016.

[32] C. H. Yang, S. C. Chen, Y. S. Tsai, R. Lu, and Y.-H. Lee, "The physical explanation of TDDB power law lifetime model through oxygen vacancy trap investigations in HKMG NMOS FinFET devices," in *Proc. IEEE IRPS*, Apr. 2017, pp. 3C-4.1–3C-4.6.

[33] B. P. Linder, J. H. Stathis, D. J. Frank, S. Lombardo, and A. Vayshenker, "Growth and scaling of oxide conduction after breakdown," in *Proc. IEEE IRPS*, Mar. 2003, pp. 402–405.

[34] S. Lombardo, J. H. Stathis, B. P. Linder, K. L. Pey, F. Palumbo, and C. H. Tung, "Dielectric breakdown mechanisms in gate oxides," *J. Appl. Phys.*, vol. 98, no. 12, 2005, Art. no. 121301.

[35] S. Y. Kim, G. Panagopoulos, C.-H. Ho, M. Katoozi, E. Cannon, and K. Roy, "A compact SPICE model for statistical post-breakdown gate current increase due to TDDB," in *Proc. IEEE IRPS*, Apr. 2013, pp. 2A.2.1–2A.2.4.

[36] D. Patra, A. K. Reza, M. Katoozi, E. H. Cannon, K. Roy, and Y. Cao, "Accelerated BTI degradation under stochastic TDDB effect," in *Proc. IEEE IRPS*, Mar. 2018, pp. 5C.5.1–5C.5.4.

[37] N. Tega *et al.*, "Increasing threshold voltage variation due to random telegraph noise in FETs as gate lengths scale to 20 nm," in *Proc. Symp. VLSI Technol.*, Jun. 2009, pp. 50–51.

[38] A. K. M. M. Islam and H. Onodera, "Worst-case performance analysis under random telegraph noise induced threshold voltage variability," in *Proc. ACM Int. Symp. Power Timing Modeling, Optim. Simulation*, Jul. 2018, pp. 140–146.

[39] K. Sonoda, K. Ishikawa, T. Eimori, and O. Tsuchiya, "Discrete dopant effects on statistical variation of random telegraph signal magnitude," *IEEE Trans. Electron Devices*, vol. 54, no. 8, pp. 1918–1925, Aug. 2007.

[40] H. Amrouch, S. Mishra, V. van Santen, S. Mahapatra, and J. Henkel, "Impact of BTI on dynamic and static power: From the physical to circuit level," in *Proc. IEEE IRPS*, Apr. 2017, pp. CR-3.1–CR-3.6.

[41] P. Weckx *et al.*, "Implications of BTI-induced time-dependent statistics on yield estimation of digital circuits," *Trans. Electron Devices*, vol. 61, no. 3, pp. 666–673, 2014.

[42] V. P. Yanambaka, S. P. Mohanty, E. Kougianos, D. Ghai, and G. Ghai, "Process variation analysis and optimization of a FinFET-based VCO," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 2, pp. 126–134, Mar. 2017.

[43] *Predictive Technology Model*. Accessed: May 2017. [Online]. Available: http://ptm.asu.edu/

[44] A. Calimera, M. Loghi, E. Macii, and M. Poncino, "Partitioned cache architectures for reduced NBTI-induced aging," in *Proc. DATE*, Mar. 2011, pp. 1–6.

[45] V. Huard *et al.*, "NBTI degradation: From transistor to SRAM arrays," in *Proc. IEEE IRPS*, May 2008, pp. 289–300.

[46] A. Bansal, R. Rao, J.-J. Kim, S. Zafar, J. H. Stathis, and C.-T. Chuang, "Impact of NBTI and PBTI in SRAM bit-cells: Relative sensitivities and guidelines for application-specific target stability/performance," in *Proc. IEEE IRPS*, Apr. 2009, pp. 745–749.

[47] S. Khan *et al.*, "Bias temperature instability analysis of FinFET based SRAM cells," in *Proc. DATE*, Mar. 2014, pp. 31.

[48] J. C. Lin, A. S. Oates, and C. H. Yu, "Time dependent Vccmin degradation of SRAM fabricated with high-k gate dielectrics," in *Proc. IEEE IRPS*, Apr. 2007, pp. 439–444.

[49] K. Kang, S. Gangwal, S. P. Park, and K. Roy, "NBTI induced performance degradation in logic and memory circuits: How effectively can we approach a reliability solution," in *Proc. Asia South Pacific Design Autom. Conf.*, Jan. 2008, pp. 726–731.

[50] P. Weckx *et al.*, "Defect-based methodology for workload-dependent circuit lifetime projections—Application to SRAM," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2013, pp. 3A.4.1–3A.4.7.

[51] E. Mintarno, V. Chandra, D. Pietromonaco, R. Aitken, and R. W. Dutton, "Workload dependent NBTI and PBTI analysis for a sub-45nm commercial microprocessor," in *Proc. IEEE IRPS*, Apr. 2013, pp. 3A.1.1–3A.1.6.

[52] D. Angot, V. Huard, M. Quoirin, X. Federspiel, S. Haendler, and M. Saliva, "The impact of high V_{th} drifts tail and real workloads on SRAM reliability," in *Proc. IEEE IRPS*, Jun. 2014, pp. CA.10.1–CA.10.6.

[53] A. Calimera, M. Loghi, E. Macii, and M. Poncino, "Dynamic indexing: Leakage-aging co-optimization for caches," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 2, pp. 251–263, Feb. 2014.

[54] A. Valero, N. Miralaei, S. Petit, J. Sahuquillo, and T. M. Jones, "On microarchitectural mechanisms for cache wearout reduction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 857–871, Mar. 2016.

[55] S. Khan and S. Hamdioui, "Trends and challenges of SRAM reliability in the nano-scale era," in *Proc. DTIS*, Mar. 2010, pp. 1–6.

[56] M. Indaco, P. Prinetto, and E. I. Vatajelu, "On the impact of process variability and aging on the reliability of emerging memories (Embedded tutorial)," in *Proc. ETS*, May 2014, pp. 1–10.

[57] V. Huard *et al.*, "Managing SRAM reliability from bitcell to library level," in *Proc. IEEE IRPS*, May 2010, pp. 655–664.

[58] J. Qin, X. Li, and J. B. Bernstein, "SRAM stability analysis considering gate oxide SBD, NBTI and HCI," in *Proc. IIRW*, Oct. 2007, pp. 33–37.

[59] N. Khoshavi, R. A. Ashraf, and R. F. Demara, "Applicability of power-gating strategies for aging mitigation of CMOS logic paths," in *Proc. MWSCAS*, Aug. 2014, pp. 929–932.

[60] A. P. Shah, N. Yadav, A. Beohar, and S. K. Vishvakarma, "On-chip adaptive body bias for reducing the impact of NBTI on 6T SRAM Cells," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 242–249, May 2018.

[61] T. Siddiqua and S. Gurumurthi, "Recovery boosting: A technique to enhance NBTI recovery in SRAM arrays," in *Proc. ISVLSI*, Jul. 2010, pp. 393–398.

[62] E. Gunadi, A. A. Sinkar, N. S. Kim, and M. H. Lipasti, "Combating aging with the colt duty cycle equalizer," in *Proc. MICRO*, Dec. 2010, pp. 103–114.

[63] J. Shin, V. Zyuban, P. Bose, and T. M. Pinkston, "A proactive wearout recovery approach for exploiting microarchitectural redundancy to extend cache SRAM lifetime," in *Proc. ISCA*, Jun. 2008, pp. 353–362.

[64] R. Rodríguez *et al.*, "The impact of gate-oxide breakdown on SRAM stability," *IEEE Electron Device Lett.*, vol. 23, no. 9, pp. 559–561, Sep. 2002.

[65] H. Wang, M. Miranda, F. Catthor, and W. Dehaene, "On the combined impact of soft and medium gate oxide breakdown and process variability on the parametric figures of SRAM components," in *Proc. IEEE MTDT*, Aug. 2006, pp. 71–76.

[66] K. V. Aadithya, A. Demir, S. Venugopalan, and J. Roychowdhury, "Accurate prediction of random telegraph noise effects in SRAMs and DRAMs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 73–86, Jan. 2013.

[67] M. Luo, R. Wang, S. Guo, J. Wang, J. Zou, and R. Huang, "Impacts of random telegraph noise (RTN) on digital circuits," *IEEE Trans. Electron Devices*, vol. 62, no. 6, pp. 1725–1732, Jun. 2015.

[68] Q. Tang and C. H. Kim, "Characterizing the impact of RTN on logic and SRAM operation using a dual ring oscillator array circuit," *IEEE J. Solid-State Circuits*, vol. 52, no. 6, pp. 1655–1663, Jun. 2017.

[69] D. Mao, S. Guo, R. Wang, M. Luo, and R. Huang, "Deep understanding of random telegraph noise (RTN) effects on SRAM stability," in *Proc. VLSI-TSA*, Apr. 2016, pp. 1–2.

[70] M. Agostinelli *et al.*, "Erratic fluctuations of sram cache vmin at the 90nm process technology node," in *IEDM Tech. Dig.*, Dec. 2005, pp. 655–658.

[71] N. Binkert *et al.*, "The Gem5 simulator," *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, May 2011.

[72] *LEON3 Processor*. Accessed: Dec. 2015. [Online]. Available: http://gaisler.com/index.php/downloads/leongrlib

[73] *MiBench Benchmark*. Accessed: Dec. 2015. [Online]. Available: http://www.eecs.umich.edu/mibench

[74] COMSOL AB, Stockholm, Sweden. *COMSOL Multiphysics V.5.2*. Accessed: Feb. 2018. [Online]. Available: http://www.comsol.com

[75] W. Ahn, S. H. Shin, C. Jiang, H. Jiang, M. A. Wahab, and M. A. Alam, "Integrated modeling of self-heating of confined geometry (FinFET, NWFET, and NSHFET) transistors and its implications for the reliability of sub-20 nm modern integrated circuits," *Microelectron. Reliab.*, vol. 81, pp. 262–273, Feb. 2018.

[76] E. Seevinck, F. J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE J. Solid-State Circuits*, vol. SSC-22, no. 5, pp. 748–754, Oct. 1987.

[77] B. Sklar and F. J. Harris, "The ABCs of linear block codes," *IEEE Signal Process. Mag.*, vol. 21, no. 4, pp. 14–35, Jul. 2004.

[78] T. Liu, C.-C. Chen, J. Wu, and L. Milor, "SRAM stability analysis for different cache configurations due to bias temperature instability and hot carrier injection," in *Proc. ICCD*, Oct. 2016, pp. 225–232.

[79] A. Ricketts, J. Singh, K. Ramakrishnan, V. Narayanan, and D. K. Pradhan, "Investigating the impact of NBTI on different power saving cache strategies," in *Proc. DATE*, Mar. 2010, pp. 592–597.