System-Level Variation-Aware Aging Simulator Using a Unified Novel Gate-Delay Model for BTI, HCI and GOBD

Taizhi Liu, Chang-Chih Chen, Soonyoung Cha, and Linda Milor

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332

Abstract

A framework is proposed to analyze system-level reliability and evaluate the lifetimes of state-of-art microprocessors considering the impact of process-voltage-temperature (PVT) variations and device wearout mechanisms bias temperature instability (BTI), hot carrier injection (HCI), and gate oxide breakdown (GOBD). This work studies not only the system performance degradation due to each wearout mechanism individually, but also the performance degradation while all these wearout mechanisms happen simultaneously. A unified gate-delay model is developed to combine PVT variations and the aging effect, and then a statistical timing engine is constructed to analyze performance degradations and system lifetimes.

1. Introduction

The device wearout mechanisms, namely BTI, HCI and GOBD, not only degrade microprocessor performances but also shorten microprocessor lifetimes. Besides transistor aging, PVT variations also lead to a large amount of variability in microprocessor performances and lifetimes. The large system complexity makes it very challenging to characterize microprocessor lifetime due to device reliability in the presence of PVT variations.

The purpose of this paper is to present a methodology to assess system-level microprocessor lifetime due to the combined effect of both PVT variations and the aging effect. Not only the performance degradation due to each wearout mechanism is studied individually, but also the performance degradation while BTI, HCI and GOBD happen simultaneously is analyzed. Because all these wearout mechanisms happen in devices at the same time, it's necessary to study the performance degradations and system lifetimes due to the combined effect of BTI, HCI and GOBD.

A unified gate-delay model is proposed to link device-level wearout models and the gate delays. The gate-delay model includes the following parameters: channel length, threshold voltage, GOBD breakdown resistances, supply voltage, temperature, input slew and Pi-model parameters (for the RC load). Among them, the threshold voltage, of each transistor combines the effect of process variation, BTI and HCI, while the GOBD breakdown resistances of each transistor represent the GOBD effect. All the included parameters have a considerable effect on gate delays. A method called multivariate adaptive regression splines (MARS) is employed to characterize the function of gate delay as a function of these parameters. MARS is well-suited for capturing essential nonlinearities and interactions in a high-dimension parameter space.

Based on the unified gate-delay models, a statistical timing engine is developed to estimate the variability of circuit-performance degradation due to the aging effect when PVT variations are present. The proposed timing engine consists of two parts: a block-based analyzer and

Fig. 1. Comparison of the critical paths between STA and PVT-aging-aware method. 2000 Monte Carlo simulations are run via SPICE on ISCAS85 benchmark c499. The same PVT variations and aging effect are applied to the SPICE simulations of both path sets.

a path-based analyzer. The block-based analyzer performs PVT-aging-aware critical-path extraction, and the path-based analyzer performs accurate circuit-delay estimation of the extracted paths.

Using the statistical timing engine, a framework of system-level aging assessment is built. Since the wearout mechanisms under study are activity, voltage (V_{DD}) and temperature dependent, the proposed framework determines the detailed electrical stress profiles, thermal profiles and IR-drop profiles of each device within the system under study. Combining these profiles and device wearout models, the statistical timing engine mentioned earlier is applied to characterize microprocessor performance degradation and assess system lifetime. Moreover, the lifetime estimates take into account the realistic use scenarios which include active, standby, and sleep modes.

When PVT variations and the aging effect are taken into account, Static Timing Analysis (STA) is not capable of identifying the real critical paths, while Statistical Static Timing Analysis (SSTA) are well used to handle variations. In this work, our framework implements block-based SSTA timing analyzer which extracts the critical paths in the presence of both PVT variations and aging effect. Fig. 1 shows an example that the critical paths extracted by STA are different from the ones extracted by the methodology proposed, and ~7% error is introduced into circuit-delay distribution using STA-extracted paths. For a large balanced circuit, using

STA-extracted critical paths results in significant inaccuracy. Also, using SPICE to run Monte Carlo simulations on critical paths is very timing-expensive, especially when the number of critical paths is large. To solve this issue, the path-based SSTA timing analyzer presented in this paper achieves SPICE-accuracy while consuming only 1% runtime compared to SPICE, making the proposed framework feasible to handle complex systems.

The rest of this abstract is organized as follows. Section 2 presents the wearout models used in this paper. Section 3 introduces the gate-delay model which integrates the impact of PVT variations, BTI, HCI, and GOBD. Section 4 shows the system-level aging assessment using a SSTA timing engine. Section 5 concludes the paper.

2. Device-level wearout models

2.1. NBTI/PBTI models

Recent experimental work has shown that the threshold voltage shift as a function of time under DC stress (t_{DC}) is best modeled with trapping/de-trapping theory [1–6]:

$$\Delta V_{tp/tn}(DC) = \phi(T, E_F)(A + B \ln(t_{DC})), \quad (1)$$

where, A, B , and ϕ are constants. ϕ is proportional to the number of available traps and is a function of temperature, T , and the Fermi level, E_F . The temperature dependence is incorporated in ϕ . We have modeled temperature with the Arrhenius relationship:

$$\phi(T, E_F) = \phi_0 g(E_F) e^{-E_a/kT}, \quad (2)$$

where E_a is the activation energy, k is a constant, and T is temperature. A frequency dependence in $\Delta V_{tp/tn}$ has not been included, since it has been shown to be relatively insignificant, especially for low frequency signals [7]. However, the duty cycle, α , impacts the shift and is incorporated as an effective Fermi level, where $E_{F,eff} = \alpha E_{F,on} + (1 - \alpha) E_{F,off}$, where $E_{F,on}$ and $E_{F,off}$ are Fermi levels when the device is on and off, respectively. The result, $g(\alpha)$, is a nonlinear function modeled of α , where $g(1) = 1$ and $g(0) = 0$ [2]. The duty cycle accounts for the time under stress, t_{stress} , and the recovery time, t_{rec} , since $\alpha = t_{stress}/(t_{stress} + t_{rec})$. Hence, overall,

$$\Delta V_{tp/tn} = \phi_0 e^{-E_a/kT} g(t_{stress}/(t_{stress} + t_{rec})) \cdot (A + B \ln(t_{stress} + t_{rec})) \quad (3)$$

where ϕ_0 is a constant. The constants were obtained from the experimental results in [8].

2.2. HCI model

Since hot electrons are generated during logic transitions, the impact of HCI is directly proportional to the switching frequency. In this paper, predictive HCI lifetime models under dynamic stress are used for long term performance-degradation simulations. The threshold voltage degradation due to HCI during stress time is modeled as [9]:

$$\Delta V_{tp/tn} = A_{HCI} (r_{trans} t_{stress} t_{trans})^n \quad (4)$$

where r_{trans} is the frequency-dependent transition rate (transitions per unit time), t_{stress} is the total time under stress/operation, t_{trans} is the transition time (rise or fall time), and A_{HCI} and n are technology-dependent

constants that relate to the inversion charge, trap generation energy, and the hot electron mean free path.

2.3. GOBD model

In order to represent soft and hard breakdown (SBD and HBD) in GOBD, the oxide breakdown resistance is modeled as a function of time with the percolation and quantum point contact (QPC) model.

In the percolation model [10], during electrical stress, the trap density in the SiO₂ increases with stress time t as a power law in the anode hole injection model. Stress is converted to a number of traps [11]:

$$N_{trap}(t, V_G) = A \exp(BV_G) t^\beta \cdot \tau_{ox} WL, \quad (5)$$

where V_G is gate voltage, A, B , and β are fitting constants, and τ_{ox}, W , and L are oxide thickness, gate width, and length, respectively. The percolation model involves placing neutral traps randomly within the oxide and analyzing the number of resistive conduction paths in a three dimensional matrix representing the oxide layer. In simulation, devices experiencing HBD and SBD are randomly selected based on their probability of stress. Next, for each device experiencing breakdown, the transistor is replaced with a resistive transistor model. For HBD, R_{HBD} is in the $k\Omega$ range. For SBD, the SBD leakage resistance is calculated with the QPC model [12],

$$R_{SBD} \cong V_G / \left[\frac{4e}{h\alpha} N \cdot \exp(-\alpha\Phi) \cdot \sinh\left(\frac{\alpha e(V_G - V_0)}{2}\right) \right], \quad (6)$$

where V_G is gate voltage, $\Phi = 3 \sim 4eV$, $V_0 = 0 \sim 0.5V$, $\alpha = 2 \sim 3eV^{-1}$, h is Plank's constant, e is the electron charge, and N is number of SBD conduction paths [12–14].

In summary, the device-level GOBD model used in this paper introduces two time-dependent resistances for each transistor, shown in Fig. 2. If the breakdown path happens between gate and drain, R_{G2D} is calculated as the breakdown resistance while R_{G2S} is adequately large to have an ignorable effect. Similarly, if the breakdown path happens between gate and source, then R_{G2S} is calculated as the breakdown resistance and R_{G2D} is adequately large.

3. Gate-level PVT-aging aware delay models

This section introduces a gate-delay model which considers PVT variations and the aging effect using a method called multivariate adaptive regression splines (MARS).

3.1. RC interconnect modelling

The effect of the resistive-capacitive (RC) interconnect is considered to ensure accurate timing estimation. For the example shown in Fig. 3(a), the interconnect network, together with the loading cells (NOR2 and INV), forms the load of the driving cell (Buffer). After modeling the loading cells (NOR2 and INV) using their input capacitances (C_{NOR2_B} and C_{INV}), the load of the driving cell is purely resistive-capacitive, shown in Fig. 3(b). Then the three-parameter Pi-model [15] shown in Fig. 3(c) is used to reduce the order of the original RC network. The Pi-model is the most popular reduced-order model to estimate the input admittance of RC interconnects. The values of C_{pi1} , R_{pi} and C_{pi2} in Pi-model shown in Fig. 3(c) are obtained by equating the first, second, and third moments of the Pi-model to

Fig. 2. Device-level GOBD model used in this paper.

Fig. 3. Modeling of RC interconnect network.

corresponding moments of the original network in Fig. 3(b). The moment generation is obtained via the modified nodal analysis (MNA) method [15].

When PVT variations are present in the loading cells (NOR2 and INV), they change the input capacitances (C_{NOR2_B} and C_{INV}) which are used to form the resistive-capacitive network in Fig. 3(b) and thus change Pi-model in Fig. 3(c). With PVT variations, C_{NOR2_B} and C_{INV} are variational. It is not satisfactory to model standard-cell input capacitances as fixed values. A first-order regression equation is used to model the input capacitances of standard cells as a function of PVT parameters, shown as in (7). More details of characterizing (7) can be found in [16].

$$C_{input_cap} = k_0 + \sum_{i=1}^n k_i X_i \quad (7)$$

Then, a statistical Pi-model is constructed using the first-order sensitivity of C_{pi1} , R_{pi} and C_{pi2} to the input capacitances of each loading cell. The statistical Pi-model is characterized for each interconnect network, that is, each interconnect network has its unique statistical Pi-model. After input capacitances of the loading cells are calculated using (7), the Pi-model parameters are quickly calculated via the statistical Pi-model. Equation (8) gives C_{pi1} as an example.

$$C_{pi1} = C_{pi1_nominal} + \sum_{i=1}^N \alpha_i (C_i - C_{i_nominal}) \quad (8)$$

where C_i denotes the input capacitance of i -th loading cell, $C_{i_nominal}$ is the nominal value of C_i , α_i is the first derivative of C_{pi1} with respect to C_i , and $C_{pi1_nominal}$ is the nominal value of C_{pi1} .

3.2. Parameters included in gate-delay model

For each transistor of a gate, the variations of channel length and threshold voltage are considered, denoted as ΔL and ΔV_{th} , respectively. The variation of ΔL comes from only process variation, while the variation of ΔV_{th} comes from process variation, BTI and HCI.

$$\Delta V_{th} = \Delta V_{th_process} + \Delta V_{th_BTI} + \Delta V_{th_HCI} \quad (9)$$

GOBD introduces two additional parameters (gate-to-source resistance, R_{G2S} , and gate-to-drain resistance, R_{G2D}) for each transistor of a gate, because the gate-oxide breakdown paths can happen at the gate-to-source or the gate-to-drain. Therefore, each transistor has four parameters (ΔL , ΔV_{th} , R_{G2S} , R_{G2D}) considered in the gate-level model. For a gate containing N transistors, there are $4N$ parameters for ΔL , ΔV_{th} , R_{G2S} and R_{G2D} .

ΔV_{DD} and ΔT are used to represent the supply voltage and temperature of a gate, respectively. It is assumed that all the transistors in the gate have the same

voltage and temperature. The three parameters (R_{pi} , C_{pi1} , C_{pi2}) in the Pi-model represent the load of a gate. Also, for each timing arc, the input slew time (*Slope*) is included. Note that the Multiple Input Switching (MIS) is not considered in this work.

In summary, for a gate containing N transistors, the $4N$ parameters (for ΔL , ΔV_{th} , R_{G2S} , R_{G2D}), together with the six parameters (ΔV_{DD} , ΔT , R_{pi} , C_{pi1} , C_{pi2} , *Slope*), construct a total of $(4N+6)$ parameters for the gate-level modelling. In the standard cell library used in this work, the value of N could be as high as 32, making $(4N+6)$ as high as 134.

3.3. The multivariate adaptive regression splines (MARS) method

The MARS method is used to model the function between gate delay/slew and the $(4N+6)$ explanatory parameters mentioned in section 3.2. The high dimension of the explanatory-parameter space, as well as the nonlinearities between gate delay/slew and the explanatory parameters, makes it very difficult to characterize this function. Traditional regression techniques like the response surface method (RSM), suffer from the disadvantage of using the same model to cover the entire parameter space. One model is insufficient to accurately estimate the gate delay (or slew time) over the whole parameter space.

MARS is well suited for high-dimensional problems while capturing essential nonlinearities and interactions. The piecewise nature of MARS allows it to split the whole high-dimension parameter space into multiple subspaces, and each subspace has a unique regression model. MARS inherently integrates all the regression models of different subspaces into one general expression using piecewise hinge functions.

A hinge function has the form of $(x - t)_+$ or $(t - x)_+$ which are shown in Figure 2. They are defined as:

$$(x - t)_+ = \begin{cases} x - t, & \text{if } x > t, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

$$(t - x)_+ = \begin{cases} t - x, & \text{if } x < t, \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

where t is a constant, called the knot. $(x - t)_+$ and $(t - x)_+$ are called a reflected pair of hinge functions.

MARS uses reflected pairs of hinge functions for each explanatory parameter X_j with knots at each observed value $(x_{1j}, x_{2j}, \dots, x_{Mj})$, where M is the number of observations.

The MARS model has the forms:

$$f(\vec{X}) = \beta_0 + \sum_{t=1}^T \beta_t h_t(\vec{X}), \quad h_t(\vec{X}) \in \mathbf{M} \quad (12)$$

where $h_t(\vec{X})$ is called a basis function, $\vec{X}$ denotes the vector of all the explanatory parameters, and set $\mathbf{M}$ denotes the collection of the basis function in the model. MARS builds a model in two phases: the forward stepwise addition and the backward stepwise deletion.

In the forward phase, MARS repeatedly adds basis functions in pairs to the model step by step. It finds the pair of basis functions that gives the maximum reduction in the sum-of-squares residue error. The process of forward stepwise addition continues until the change in

Table 1

The corners of the parameters in gate-delay model

Var.	Corners	Var.	Corners
ΔL_p	[-30%,30%]	ΔL_n	[-30%,30%]
ΔV_{th_p}	[-30%,30%]	ΔV_{th_n}	[-30%,30%]
ΔVDD	[-10%,10%]	$\Delta T(^{\circ}C)$	[-50,50]
$Slope$	[10ps,3ns]	$R_{pi}(\Omega)$	[1,1000]
$C_{pi1}(pF)$	[0.1,10]	$C_{pi2}(pF)$	[1,100]
$R_{G2S}(\Omega)$	[43k,15M]	$R_{G2D}(\Omega)$	[43k,15M]

Table 2

MARS models on representative standard cells

Cell	(4N+6)	Time(s)	Error	
			Mean	S.D.
INVX1	14	861	0.41%	2.54%
NOR2X1	22	1281	0.23%	2.67%
CLKBUF3	134	6577	0.17%	2.86%
XOR2X1	54	3112	0.20%	2.32%

residual error is smaller than a threshold or until the maximum number of terms is reached. The backward stepwise deletion prunes the model to avoid overfitting. More details can be found in [17].

3.4. Gate-level delay model

SPICE simulations are used as training data to train the MARS model. Design of the experiments for SPICE simulation is using the mixture of central composite design [18] and random samples within the corners. The corners of explanatory parameters which is used for central composite design is shown in Table 1. The implementation of MARS is from a Matlab toolbox called ARESLab [19]. Every timing arc in every cell (a total of 247 standard cells in our library) is characterized. Table 2 presents the results of some representative cells. The “4N+6” column denotes the number of parameters in the gate-delay model, and the “Time(s)” column denotes the run time to characterize the MARS gate-delay model. The two columns (mean and S.D.) in “error” denote the average and standard deviation of the errors between MARS and SPICE, respectively. With these gate-delay models, the device-level wearout models are converted to gate-delay degradation which is further used to facilitate system-level aging assessment.

The interconnect-delay model is also obtained using MARS. A method called the stable two-pole (S2P) approximation [20] was employed to get the reduced-order model of the transfer function of the original RC network. The three parameters from this reduced-order model, together with ΔVDD , ΔT and $Slope$, are used in the interconnect-delay MARS model.

4. System-level aging assessment framework

4.1. Extraction of the stress, thermal, IR-drop and process profiles

The wearout mechanisms under study are activity, voltage (VDD) and temperature dependent. The degradation of system performance is also directly dependent on the thermal and the IR-drop profile because temperature and VDD directly impact circuit timing. Therefore, this paper proposes the framework shown in Fig. 4 to extract the spatial and temporal electrical stress, IR-drop, the thermal profile of a system.

Running RTL or SPICE simulations of a complete microprocessor to extract the activity profile of each net is not feasible in most cases, since it may take a few

Fig. 4. The framework of our aging simulator.

Algorithm: Statistical Timing Engine

Input: circuit netlist, RC parasitics (.spef), process-variation spatial correlation profile, BTI ΔV_{th} profile, HCI ΔV_{th} profile, GOBD R_{G2S}/R_{G2D} profile, thermal profile, IR-drop profile
Output: Circuit-delay distribution

Block-based timing analyzer

1. generate N Monte Carlo samples for the circuits according to process-variation spatial correlation profile.
2. abstract timing graph from circuit netlist, and apply (ΔL , ΔV_{th} , R_{G2S} , R_{G2D})
3. for $i=1$ to N //for each Monte Carlo sample
 - 3.1. apply to each gate the corresponding values of ΔL , ΔV_{th} , R_{G2S} , R_{G2D} , T and etc.
 - 3.2. do forward traversal to evaluate the edges (gate and wire delay) and propagate the arrival time to Primary Output/DFP
 - 3.3. do backward traversal to extract the first 3 critical paths using PERT algorithm.

Path-based timing analyzer

4. end for
5. get the critical-path collection: $P = \{CP_1, CP_2, \dots, CP_m\}$
 // m is up to $3*N$, because some paths are repetitive
6. for $i=1$ to N //for each Monte Carlo sample
 - 6.1. calculate the path delays for each critical path in P as $\{PD_1^i, PD_2^i, \dots, PD_m^i\}$
 - 6.2. get the maximum path delay $D^i = \max\{PD_1^i, PD_2^i, \dots, PD_m^i\}$
7. end for
8. circuit-delay distribution from $\{D^1, D^2, \dots, D^N\}$

Fig. 5. The framework of the PVT-aging-aware timing engine

Fig. 6. Abstraction of a timing graph from a combinational circuit.

Table 3

Test Results on our timing engine compared to SPICE (with stress time = 10 years)

Circuits	Cell Number	Time Saved	SPICE(ps)		Our Engine(ps)	
			mean	S.D.	mean	S.D.
TA1	21,725	99.1%	645	79	652	82
TA2	78,623	99.5%	1231	130	1243	136
leon2	454,489	99.0%	1960	412	2005	427
netcard	425,264	99.1%	1842	393	1879	412
leon3mp	330,993	98.8%	1211	368	1256	375
leon3-avnet-3s1500	542,081	99.2%	1981	432	2022	448

months to finish simulating a single benchmark. On the other hand, simulating microprocessors with standard benchmarks on an FPGA takes only a few minutes [21]. Our framework shown in Fig. 4 provides an efficient way to acquire electrical, thermal and IR-drop profiles for any digital system for use in system-level reliability analysis.

For activity tracking, the hardware netlists were loaded onto an FPGA for emulation [22–23], and monitors were placed at the I/O ports, which track both the state probabilities and the toggle rates at the ports during application runtime. A set of standard benchmarks [24] were used as the applications. The I/O activities and the netlists were then used for activity propagation to each internal net for a complete

stress/transition probability profile of the internal nodes of the microprocessors.

The thermal and IR-drop profiles throughout the microprocessor are also extracted. The netlist was used for layout generation. 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 simulator and the thermal simulator, respectively.

For process variations, the proposed methodology includes inter-die, intra-die, and intra-gate variations supporting any kind of distribution and any correlation profile between different parameters. The process variations of channel length (ΔL) and threshold voltage (ΔV_{th}) for each transistor are considered. Because BTI and HCI also impact the threshold voltage, the threshold voltage at each transistor is a combination of process variation, BTI and HCI, while process variation contributes all the variations of ΔL .

The electrical stress, thermal, IR-drop and process profiles and RC parasitics, together with the device-level wearout models, generate all the parameters of each gate to calculate the gate delays. All the information is fed to a PVT-aging aware timing engine for the analysis of performance degradation. The timing engine generates the delay degradations under a variety of stress time, and calculates the lifetime according to the specified operating frequency. After a set of Monte Carlo samples are run, the lifetime distribution is obtained.

4.2. PVT-aging aware timing engine

The PVT-aging aware timing engine consists of two parts, block-based analyzer and a path-based analyzer. The block-based timing analyzer extracts real critical paths considering PVT variations and the aging effect; the path-based timing analyzer performs timing analysis fed by the critical paths extracted from the block-based analyzer. Our timing engine has two features: it performs PVT-aging aware critical-path extraction, and it achieves SPICE accuracy, while consuming only 1% runtime.

Fig. 5 gives the algorithm of the timing engine. Both the block-based and path-based timing analyzer are based on Monte Carlo analysis. The block-based timing analyzer first abstracts a timing graph from the gate-level netlist, shown in Fig. 6. The nodes of the timing graph represent primary input/outputs and gate input/output pins. Its edges represent the timing elements of the circuit, namely, the gate input-pin–output-pin delay and the interconnect delay. The weight on these edges is the delay of the corresponding timing elements, which is calculated using the unified gate-level MARS models. After a forward traversal of the timing graph, the arrival times at primary outputs and D inputs of flip-flops are obtained. By a backward traversal, the critical paths can be extracted using the PERT algorithm [25]. For each Monte Carlo sample, the process of delay evaluation, forward traversal and backward traversal is repeated. With a number of samples, a set of critical paths is obtained in the presence of PVT variations and the aging effect.

Fig. 7. The temperature (part (a)) and VDD (part (b)) distribution of the microprocessor.

Fig. 8. The distributions of the stress probability (part (a)) and the transition rate (part (b)) for the microprocessor while running a standard benchmark.

Fig. 9. The lifetime distributions of the microprocessor due to BTI, HCI, GOBD and the combined effect.

The path-based timing analyzer is fed by a set of paths. The delay of each path is calculated by accumulating the gate delay and interconnect delay along the path using the MARS delay models. Then the maximum value of the path delays is taken as the circuit delay. Similarly, after a number of Monte Carlo runs, the circuit-delay distribution is obtained.

Our timing engine exploits the advantages of both the block-based timing analyzer and the path-based timing analyzer, by using the block-based analyzer to extract accurate critical paths and using path-based analyzer to determine an accurate timing distribution. The efficiency and accuracy of our timing engine has been verified using several industrial designs from the IWLS2005 benchmarks [26], as the results are shown in Table 3. The computational complexity of our timing engine is $O(MN)$, where M is the number of gates and N is the number of samples. The results in Table 3 show the comparison of the mean and standard deviation (S.D.) of the circuit-delay distribution between SPICE and our timing engine. Our timing engine achieves SPICE accuracy while consuming 1% runtime of SPICE, and hence facilitates its usage in the system-level aging simulator which is introduced in Section 4.3.

4.3 System-Level Aging Simulator

The core of this aging simulator is the PVT-aging aware timing engine as shown in Fig. 4. The timing engine is fed by a set of profiles including process variation, thermal, IR-drop, RC parasitics, BTI, HCI and GOBD. The generation of these profiles are discussed in Section 4.1.

The system-level aging simulator is constructed by analyzing the performance (circuit-delay) degradation at different stress times. When the circuit-delay degrades beyond the clock period, the system is seen as a failure and the lifetime is thus obtained. Our system-level aging simulator has been applied to the LEON3 microprocessor [27], which consists of logic units (IU, MUL, DIV and MMU) and storage blocks (RF, D-Cache, I-Cache, Dtags and Itags).

Fig. 7 shows the distributions of the state probability and the transition rate in part (a) and (b), respectively, when LEON3 is running a set of standard benchmarks. Fig. 8 shows the distributions of the state probability and the transition rate in part (a) and (b), respectively. Fig. 9 shows the comparison of the lifetime distributions due to BTI, HCI, GOBD individually and simultaneously. It is seen that GOBD is generally the most critical mechanism of the three, but for some samples BTI is dominant. Besides, when all these three mechanisms happen simultaneously, the lifetime is shorter than the lifetimes of analyzing each mechanism alone. It is concluded that the BTI and HCI also have some effect on the circuit lifetimes even they are not dominant.

5. Conclusion

A system-level aging simulator which evaluates the microprocessor lifetimes due to the combined effect of BTI, HCI and GOBD was proposed. The proposed work has the feature of extracting the critical paths considering PVT variations and aging effect. It can be extended for yield enhancement [28].

References

- [1] J.B. Velamala, K.B. Sutaria, T. Sato, and Y. Cao, "Aging statistics based on trapping/detrapping: Silicon evidence, modeling and long-term prediction," *Proc. IRPS*, 2012, pp. 2F.2.1-2F.2.5.
- [2] G.I. Wirth, R. da Silva, and B. Kaczer, "Statistical model for MOSFET bias temperature instability component due to charge

- trapping," *IEEE Trans. Electron Devices*, 2011, vol. 58, no. 8, pp. 2743-2751.
- [3] S. Cha, C.-C. Chen and L. Milor, "System-level estimation of threshold voltage degradation due to NBTI with I/O measurements," *Proc. IRPS*, pp. PR.1.1-PR.1.7, 2014.
- [4] S. Cha, C.-C. Chen, T. Liu and L. Milor, "Extraction of threshold voltage degradation modeling due to Negative Bias Temperature Instability in circuits with I/O measurements," *IEEE VTS*, pp. 1-6, 2014.
- [5] C.-C. Chen, F. Ahmed, L. Milor, "Impact of NBTI/PBTI on SRAMs within microprocessor systems: Modeling, simulation, and analysis," *Microelectronics Reliability*, vol. 53, pp. 1183-1188, 2013.
- [6] S. Cha, C.-C. Chen and L. Milor, "Frontend wearout modeling from device to system with power/ground signature analysis," *IEEE IIRW*, 2014.
- [7] R. Fernandez, B. Kaczer, A. Nackaerts, S. Demuynck, R. Rodriguez, M. Nafria, and G. Groeseneken, "AC NBTI studies in the 1 Hz - 2 GHz range on dedicated on-chip CMOS circuit," *Proc. Int. Electron Devices Meeting*, 2006.
- [8] S. Zafar, Y.H. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stathis, A. Callegari, and M. Chudzik, "A comparative study of NBTI and PBTI (charge trapping) in SiO₂/HfO₂ stacks with FUSI, TiN, Re Gates," *Proc. Symp. VLSI Tech.*, 2006, pp. 23-25.
- [9] W. Wang, *et al.*, "Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology," *IEEE Trans. Device and Materials Reliability*, 2007, vol. 7, no. 4, pp. 509-517.
- [10] J. H. Stathis, "Percolation models for gate oxide breakdown," *J. Appl. Phys.*, 1999, vol. 86, no. 10, pp. 5757-5766.
- [11] R. Degraeve, *et al.* "A new model for the field dependence of intrinsic and extrinsic time-dependent dielectric breakdown," *IEEE Trans. Electron Devices*, 1998, vol. 45, no. 2, pp. 472-481.
- [12] S. Cha, *et al.* "Gate oxide breakdown parameter extraction with ground and power supply signature measurements," *Proc. DCIS*, 2014, pp. 1-6.
- [13] C.-C. Chen, S. Cha and L. Milor, "System-level modeling of microprocessor reliability degradation due to TDDB," *Proc. DCIS*, 2014.
- [14] E. Miranda, *et al.* "Analytic modeling of leakage current through multiple breakdown paths in SiO₂ films," *Proc. IRPS*, 2001, pp. 367-379.
- [15] M. Celik, L. Pileggi, and A. Odabasioglu, *IC Interconnect Analysis*. New York: Springer Science & Business Media, 2002.
- [16] T. Liu, C.-C. Chen and L. Milor, "Accurate standard cell characterization and statistical timing analysis using multivariate adaptive regression splines," *Proc. ISQED*, 2015, pp. 272-279.
- [17] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1-67, March 1991.
- [18] G.E.P. Box, W.G. Hunter, and J.S. Hunger, *Statistics for Experimenters*, 2nd Edition. New York: John Wiley and Sons, 2005.
- [19] G. Jekabsons, ARESLab: Adaptive Regression Splines toolbox for Matlab/Octave, 2011, available at <http://www.cs.rtu.lv/jekabsons/>.
- [20] E. Acar, *et al.* "S2P: A stable 2-pole RC delay and coupling noise metric," in *Proc. Great Lakes Symp. on VLSI*, 1999.
- [21] C.-C. Chen, *et al.*, "System-level modeling of microprocessor reliability degradation due to BTI and HCI," *IEEE IRPS*, pp. CA.8.1-CA.8.9, 2014.
- [22] C.-C. Chen, F. Ahmed, and L. Milor, "A Comparative Study of Wearout Mechanisms in State-of-Art Microprocessors," *Proc. IEEE ICCD*, pp. 271-276, 2012.
- [23] C.-C. Chen, and L. Milor. "System-level modeling and reliability analysis of microprocessor systems." *IEEE International Workshop Advances in Sensors & Interfaces*, pp. 178-183, 2013.
- [24] Mibench benchmark: <http://www.eecs.umich.edu/mibench>.
- [25] T. Kirkpatrick and N. Clark, "PERT as an aid to logic design," *IBM J. Res. Develop.*, vol. 10, no. 2, pp. 135-141, Jun. 1966.
- [26] Available: <http://iwls.org/iwls2005/benchmarks.html>.
- [27] LEON3 processor: www.gaisler.com
- [28] Jing Li, Haixin Liu, Sayeef Salahuddin and Kaushik Roy, "Variation-Tolerant Spin-Torque Transfer (STT) MRAM Array for Yield Enhancement", *IEEE Custom Integrated Circuits Conference (CICC)*, 2008.