SYSTEM-LEVEL MODELING AND RELIABILITY ANALYSIS OF MICROPROCESSOR SYSTEMS

A Dissertation
Presented to
The Academic Faculty

by

Chang-Chih Chen

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2014

SYSTEM-LEVEL MODELING AND RELIABILITY ANALYSIS OF MICROPROCESSOR SYSTEMS

Approved by:

Dr. Linda Milor, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Abhijit Chatterjee
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David Keezer
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Hyesoon Kim
College of Computing
Georgia Institute of Technology

Dr. Azad Naeemi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: August 19, 2014

Dedicated to my parents, Lo-Wen Chen and Shu-Yuan Tang, for their endless love.

ACKNOWLEDGEMENTS

I would like to extend special thanks to my advisor, Professor Linda S. Milor, for her guidance and advice during my Ph.D. study. I would also like to thank Professor David Keezer and Professor Azad Naeemi for their helpful suggestions. I would like to express my thanks to Professor Abhijit Chatterjee and Professor Hyesoon Kim, both of whom have agreed to serve on my dissertation committee. I am also grateful to Professor Sung Kyu Lim.

I would like to express my deepest thanks to Dr. Dae Hyun Kim, for developing the layout extractor for the reliability simulator from the GTCAD lab. I am also grateful to Dr. Muhammad Bashir for his collaboration.

I thank all the lab members, Dr. Fahad Ahmed, Soonyoung Cha, Taizhi Liu and Woongrae Kim for their collaboration, and valuable comments and feedback.

I would also like to extend special thanks to my best friend, Anshuman Goswami, who have always supported and motivated me.

I am particularly thankful to my parents, Lo-Wen Chen and Shu-Yuan Tang, who have always been on my side, for their love and encouragement throughout my life. I also thank my grandfather, grandmothers, brother, Hsuan-Chih Chen, and sister, Jou-Chou Chen, for their love and support. Last but not least, I would like to thank all the professors, teachers, families, and friends who guided me to become the person that I am today.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AC	Alternating Current
BTDDB	Backend Time-Dependent Dielectric Breakdown
BTI	Bias Temperature Instability
Cu	Copper
DC	Direct Current
D-Cache	Data Cache
DIV	Divider
Dtags	Data tags
EM	Electromigration
FPGA	Field-Programmable Gate Array
GOBD	Gate Oxide Breakdown
HBD	Hard Breakdown
HCI	Hot Carrier Injection
I/O	Input/Output
IC	Integrated Circuit
I-Cache	Instruction Cache
IP	Intellectual Property
Itags	Instruction tags
IU	Integer Unit
MMU	Memory Management Unit
MTTF	Mean-Time-To-Failure

MUL	Multiplier
NBTI	Negative Bias Temperature Instability
NMOS	N-Channel MOSFET
PBTI	Positive Bias Temperature Instability
PM	Percolation Model
PMOS	P-Channel MOSFET
QPC	Quantum Point Contact
RC	Resistance and Capacitance
RF	Register File
RISC	Reduced Instruction Set Computing
SBD	Soft Breakdown
SILC	Stress Induced Leakage Current
SIV	Stress-Induced Voiding
SNM	Static Noise Margin
SPICE	Simulation Program with Integrated Circuit Emphasis
SRAM	*Static Random-Access Memory*
STA	Static Timing Analysis
TF	Time-to-Failure

SUMMARY

The object of the research is to develop a methodology to assess microprocessor lifetimes for a variety of wearout mechanisms by building the link between the device-level wearout models and the system level. This research has focused on seven critical wearout mechanisms, namely negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), hot carrier injection (HCI), transistor gate oxide breakdown (GOBD), electromigration (EM), stress-induced voiding (SIV) and backend time-dependent dielectric breakdown (BTDDB), and has demonstrated the feasibility of the proposed methodology by presenting results from a lifetime simulator based on the proposed methodology.

We have developed an emulation framework for each of these failure mechanisms. It uses an FPGA based platform to determine the activity and state profiles. The activity and state profiles are needed to determine the thermal profiles and electrical stress of each feature in a system [1]-[8]. Taking into account the detailed thermal and electrical stress profiles, a methodology was developed to accurately assess state-of-art microprocessor reliability due to different wearout mechanisms. Backend wearout mechanisms are handled differently than frontend wearout mechanisms.

Analysis of lifetime due to the backend wearout mechanisms (BTDDB, EM, SIV) is based on layout analysis, layout feature extraction, where the wearout of each feature is computed and the distributions are combined analytically to estimate the lifetime of the full system [1]-[5],[9],[10].

Frontend wearout mechanisms (NBTI, PBTI, GOBD, HCI) first degrade transistor characteristics as a function of stress, which in turn degrades circuit performances. Hence, analysis of lifetime due to frontend wearout mechansms must take into account the use conditions and the circuit performance requirements [5]-[8]. Moreover, memory performances are different than logic performances and must be handled appropriately, taking into account the memory specifications, such as the static noise margin and minimum Vdd retention voltage [6]-[8].

This work presents a way to establish the link between the device-level wearout models and the architecture level. Combining the wearout models, the thermal profiles, and the electrical stress profiles, this work provides insight into lifetime-limiting wearout mechanisms, along with the reliability-critical microprocessor functional units for a system while taking into account a variety of use scenarios, composed of a fraction of time in operation, a fraction of time in standby, and a fraction of time when the system is off. This enables circuit designers to know if their designs will achieve an adequate lifetime and further make any updates in the designs to enhance reliability prior to committing the designs to manufacture.

CHAPTER 1

INTRODUCTION

Although constant technology scaling has resulted in considerable benefits, including smaller device dimensions, higher operating temperatures and electric fields have also contributed to faster device and interconnect aging due to wearout. Not only does this result in the shortening of microprocessor lifetimes, it leads to faster wearout resultant performance degradation with operating time. Microprocessor lifetime is a function of both device and backend wearout.

The analysis of frontend mechanisms is different than backend mechanisms. Backend mechanisms result in open and short circuits, which result in system failure directly, and hence it is sufficient to model the time-to-failure of components of the system and to combine them statistically. Frontend wearout mechanisms, on the other hand, cause a gradual weakening of the devices. The weakening is both random and a function of stress and temperature. However, unlike backend mechanisms, the relationship between the degradation and the circuit performances must be taken into account to determine the lifetime distribution.

Device lifetime is a function of two kinds of stress: electrical and thermal. An increase in either of the two results in decreased device reliability. The increase in device densities has been achieved through reduction in device dimensions, which means that the devices undergo increased electrical stresses during their lifetime. The resulting increase in operating frequency, as well as device densities, had led to greater thermal stress, which also increases with each new generation. A decrease in device reliability

and the increase in system complexity translate into systems whose lifetime characterization is both challenging due to the large number of devices that degrade simultaneously in modern systems and extremely critical because each device fails more quickly than in previous technologies. This work considers frontend wearout due to negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), hot carrier injection (HCI), and transistor gate oxide breakdown (GOBD).

Besides wearout due to devices, each technology generation reduces the interconnect dimensions without always reducing the supply voltage in proportion, resulting in higher electric fields within the backend dielectric and within the metal lines, increasing wearout in the backend geometries. At the same time, as the dielectric constant (k) decreases to reduce parasitics, the porosity of materials must increase, at the possible cost of increasing the vulnerability of materials to breakdown. Additionally, the faster operating frequencies of processors result in decreased interconnect reliability, due to increases in both electrical current and operating temperature, increasing the risk of failure of chips due to backend wearout for the newer technology nodes. This work considers backend wearout due to electromigration (EM), stress-induced voiding (SIV) and backend time-dependent dielectric breakdown (BTDDB).

The physics describing IC failure mechanisms both in the frontend and in the backend has matured as a result of years of refinement to existing theories. However, the extension of these models to large and complex microprocessor systems has not proven to be straightforward and is complex. Microprocessor system reliability analysis requires techniques to extend the results gathered from small test structures to large complex microprocessors. Such an endeavor requires methods to manage the deluge of data that

comes with analyzing large numbers of complex layouts and devices degrading at different rates.

The purpose of this research is to present a methodology to assess microprocessor lifetimes and circuit performances due to NBTI, PBTI, HCI, GOBD, BTDDB, EM and SIV by developing the link between the device-level wearout models and the architecture level while taking into account realistic use scenarios [11]. This enables a designer to make any updates in the design to enhance reliability prior to committing a design to manufacture.

Since the wearout mechanisms being studied are activity and temperature dependent, the proposed framework determines the detailed thermal profiles of the systems under study, as well as the electrical stress of each net/device in the systems by running a variety of standard benchmarks. Microprocessors contain both logic and SRAM components. Hence, both types of blocks are considered in this work. Backend wearout mechanisms are handled differently than frontend wearout mechanisms.

Backend wearout mechanisms impact circuits by causing short circuits (for BTDDB) and open circuits (for EM and SIV). It is assumed that these open and short circuit failures cause the system to fail, except when the failure happens in a memory block utilizing error correction codes and/or reconfiguration through redundancy. Hence, backend wearout models involve combining the time-to-failure distributions of large numbers of components and the determination of whether a component failure causes a system to fail is not required. Analysis of lifetime due to the backend wearout mechanisms is based on layout analysis, layout feature extraction, where the wearout of

each feature is computed and the distributions are combined analytically to estimate the lifetime of the full system [1]-[5],[9],[10], as illustrated in Figure 1.1.

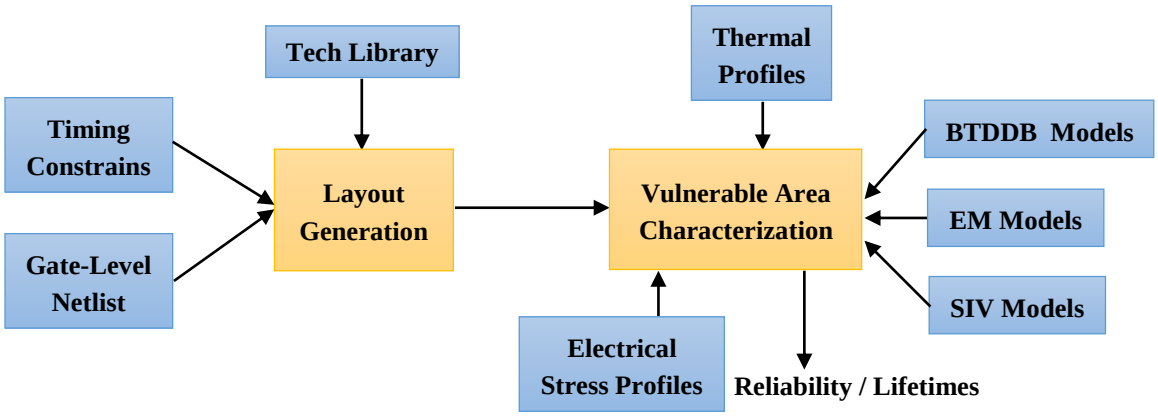


Figure 1.1: The flow of system-level modeling for backend wearout mechanisms.

Frontend wearout mechanisms, namely NBTI, PBTI, HCI, and GOBD, result in threshold voltage drifts and gate current leakage that impact circuit timing for logic blocks and SRAM performances. When studying logic blocks, we combine the electrical stress profiles, thermal profiles and device-level models, and apply statistical timing analysis (incorporating process variations) to identify the critical paths of the microprocessors and to characterize microprocessor performance degradation due to NBTI, PBTI, HCI and GOBD, as illustrated in Figure 1.2. Similarly, DC noise margins of SRAM cells are also analyzed due to NBTI, PBTI, HCI and GOBD degradation.

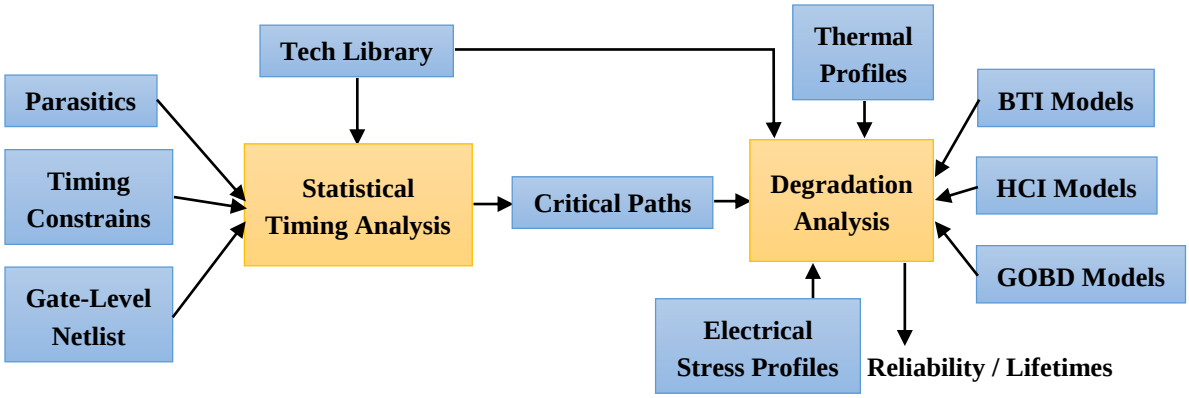


Figure 1.2: The flow of system-level modeling for frontend wearout mechanisms.

The impact of NBTI and PBTI on circuits has been previously studied with the older reaction diffusion theory ([12]-[17] for logic and [18],[19] for SRAMs) and the current trapping/detrapping theory ([20],[21] for logic and [22] for memory), which is also implemented in this work. Only simple ring oscillators are considered in [20], while providing evidence to validate trapping/detrapping theory over reaction diffusion theory.

In this work, as in prior work [23], oxide breakdown is modeled by inserting a gate-to-source resistance (R_{G2S}) or gate-to-drain resistance (R_{G2D}) in a target gate in order to create the current leakage path in the circuit. A percolation model is used to count the number of conduction paths and the time to soft breakdown (SBD) and hard breakdown (HBD) in the thin oxide layer, and a quantum point contact (QPC) model is used to calculate the SBD and HBD resistances.

Timing analysis is implemented in [21], including the updating of path selection throughout the aging process. However, prior work has involved smaller circuits and assumptions about stress distributions for each device [21],[22]. In this work we have used emulation to handle large systems running actual benchmarks to determine the actual activity of circuits and memory cells while running benchmarks. The results from

emulation are used to update timing analysis and analysis of memory performances based on actual usage patterns.

This work not only accounts for activity and temperature, but also accounts for the fact that processors are not in operation at all times. Realistic use conditions include operation modes, standby, and periods of time when the processor is turned off, as illustrated in Figure 1.3. This research presents a method to take these use scenarios into account.

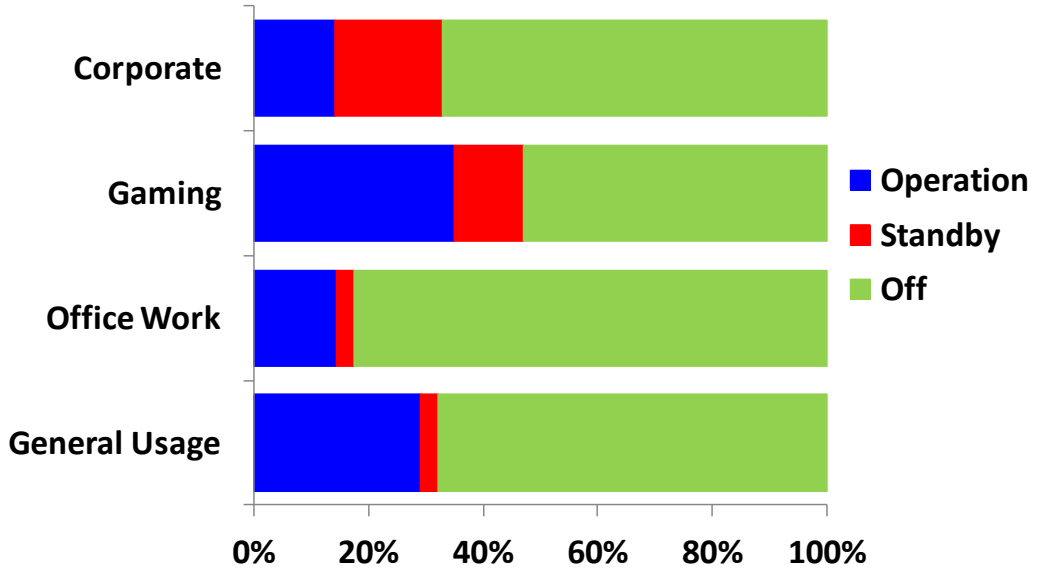


Figure 1.3: The use scenarios provided by Intel are shown [11].

The rest of the thesis is organized as follows. Chapter 2 gives a brief overview of the related work and recent trends. Chapter 3 presents the device-level wearout models we have used in this research. Chapter 4 gives the overview of our system-level aging assessment framework. The methodology to determine model parameters through FPGA emulation is described. In Chapter 5, we study the lifetimes for the systems from our

simulator and present a comparison based on our results for backend wearout mechanisms. Chapter 6 describes our methodology to evaluate performance degradation of a microprocessor due to frontend wearout mechanisms and presents the degradation and lifetime results for logic blocks of the microprocessors. We also present analysis of SRAM noise margins under BTI, HCI and GOBD degradation. Chapter 7 concludes this work.

CHAPTER 2

BACKGROUND

Aggressive technology scaling, resulting in higher operating temperatures, electric fields, and smaller device dimensions, has contributed to faster device aging. Historically, the major causes of wearout in the field have been electromigration (EM), gate oxide breakdown (GOBD), and hot carrier injection (HCI) [24]. EM [25]-[28] refers to the dislocation of metal atoms caused by momentum imparted by electrical current in interconnects and vias. The dislocation of metal atoms further causes interconnects to have increased resistance over time. The increase in resistance is design dependent, since it is a function of current density and temperature. Failure happens at joints between interconnect lines and vias, most often under the vias, where a void can form. Specifically, vias are damaged by downstream electron flow, from the via to the metal below it. GOBD [29]-[35] is detected by leakage currents through gate oxides. These leakage currents are a cumulative function of the local electric field over time and temperature. Failures in the gate oxide are caused by local thinning of the oxide due to lattice problems, such as the dislocation of an atom or the generation of traps. HCI [36]-[40] degrades device saturation current, threshold voltage, and the maximum transconductance over time, and it is due to velocity saturation effects and the reduction of charged interface states. Historically, HCI was only a major concern for NMOS devices, with PMOS devices showing comparatively negligible degradation because (a) holes have a smaller impact ionization rate and (b) holes face a higher S_i–S_iO_2 barrier than electrons. However, subsequent reports have revealed that HCI effects on PMOS

devices is also observed [41]. The rate of degradation due to HCI is sensitive to operating conditions.

More recently, because of the introduction of new materials (copper, low-k intra and inter-layer dielectrics, high-k gate dielectrics), the increase in the number of interconnect layers with smaller geometries and higher current densities, and the concomitant increase in on-chip temperatures, new failure mechanisms have emerged, including bias temperature instability in PMOS and NMOS transistors, backend time-dependent dielectric breakdown (BTDDB), and stress-induced voiding (SIV). Negative bias temperature instability (NBTI) in PMOS devices [42]-[45] is caused by the generation of interface traps under high temperature and negative gate bias and results in shifts in device parameters, such as threshold voltage, transconductance, device mobility, etc., but is generally identified by shifts in the threshold voltage [43]. Positive bias temperature instability (PBTI) has the same effect on NMOS transistors. Failures in the backend dielectric [46]-[53] are due to the alignment of trap sites which provide a low impedance path through the oxide that enables copper drift. Breakdown is detected by leakage current through the oxide. Finally, the impact of stress migration is high resistivity and opens at via sites. Stress migration is a function of interconnect geometry and is caused by the directionally biased motion of atoms in interconnects due to mechanical stress caused by thermal mismatch between metal and dielectric materials [54]-[56].

All of these wearout mechanisms cause parametric variation as a function of time. They degrade interconnect resistance, device saturation currents and/or threshold voltages, and increase the current through thin and thick oxides as a function of operating

conditions and temperature. All of these wearout mechanisms are accelerated with temperature, depend on thermal cycles (which can induce recovery for some mechanisms), and are exacerbated by thermomechanical mismatch of materials (which is degrading with the use of lower-k dielectrics in the backend).

System-level reliability analysis under realistic workloads has been studied for many years. The existing state-of-the-art is summarized in [57],[58]. In both approaches, the system is assumed to be a series combination of the components for reliability estimates, where if any component fails due to any wearout mechansism, the system fails.

In order to evaluate system-level behavior and insure reliable system operation, the gap between the established device-level wearout models and system behavior at the architecture level need to be bridged. In [59], a so-called RAMP model which conducts dynamic reliability management for analyzing microprocessor lifetime and reliability was proposed. The model assumes the device density throughout the chip is uniform and each device is identically vulnerable to failure mechanism. Later, the work proposed in [60] introduces a structure-aware model that takes into account the vulnerability of basic structures of the microarchitecture to different failure mechanisms. For the approaches to analyze system level reliability in [57]-[60], an exponential failure rate distribution is assumed. In this case the mean-time-to-failure of the chip, $MTTF_{chip}$, is a combination of the mean-time-to-failures due to each of the wearout mechanisms, $MTTF_i$. It is assumed that a *MTTF* can be computed for each wearout mechanism. Under these conditions,

$$MTTF_{chip} = 1/\sum_i (1/MTTF_i)\ . \tag{2.1}$$

This distribution does not take into account randomness in the rates of wearout for the same failure mechanism for multiple components undergoing the same stress. Moreover,

it is unrealistic to use a *MTTF* to represent chip lifetime for each wearout mechanism since a chip is composed of a large number of elements, all failing at different rates, based on their temperature, electrical stress, and geometry. To account for variation in the wearout rate, the standard distribution used in industry is the two-parameter Weibull distribution, described by a characteristic lifetime, η, and a shape parameter, β. When probabilities of failure are combined with realistic failure rate distributions, as in [58], the formulas for the time-to-failure for a chip are less straightforward. Specifically, as illustrated in [61], for each wearout mechanism, *i*, let the characteristic lifetimes and shape parameters be η_i and β_i, respectively. The characteristic lifetime of the chip does not have a closed form solution, unless β_i is constant for all wearout mechanisms. Otherwise, the characteristic lifetime, η_{chip}, is the solution of [61]-[63]

$$1 = \sum_i (\eta_{chip}/\eta_i)^{\beta_i}. \tag{2.2}$$

The shape parameter for the chip is

$$\beta_{chip} = \sum_i \beta_i (\eta_{chip}/\eta_i)^{\beta_i}\ . \tag{2.3}$$

The lifetime at probability point, *P*, is

$$-\ln(1-P) = \sum_{i=1}^{n} (t/\eta_i)^{\beta_i}\ . \tag{2.4}$$

All prior work begins with device-level models of each wearout mechanism. For instance, NBTI is a function of the build-up of interface traps, which increases as a power low function of the time under stress. When stress is removed there is a recovery, which reduced the interface traps as a function of time. The number of interface traps translates directly into a shift in the threshold voltage. However, prior work does not say much about how much of a threshold voltage shift can be tolerated by the system, and when a specific threshold voltage shift results in hard failure. The hard breakdown point is a

function of the circuit design and type of component. Similarly, GOBD is a function of stress of the oxide, which causes the formation of traps in the oxide. When the traps increase to a critical level, the leakage current through the oxide increases. When the leakage current exceeds a limit, hard breakdown occurs, and the circuit no longer functions correctly. The limit at which hard failure occurs is a function of the type of circuit and circuit specifications.

The long-term threshold voltage drifts induced by NBTI, PBTI and GOBD degrade SRAM cell stability, margin, and performance, and lead to eventual functional failure. During SRAM design, it is important to build in design margins to achieve an adequate lifetime [64],[65]. As this has become more challenging, several authors have proposed methods to improve SRAM reliability in the presence of NBTI/PBTI and GOBD degradation. These approaches include circuitry that periodically flips the data in an SRAM cell to reduce failure rates [66], the use of redundancy [67]-[69], error correcting codes [70],[71], and both [72]. Evaluation of these methods requires a model of cell stress. Assumptions are usually made about the stress distribution among cells. This is because characterizing each SRAM cell based on actual operating conditions is not straightforward.

All prior work relies on system-level benchmarks, and realistic workload models. But, little is said about the simulation method and limitations. In [73], a system-level reliability simulator was developed that includes EM, SIV, GOBD, and thermal cycling based on process-level models. However, the implementation is limited to 50,000 or fewer devices. Benchmarks for architicture evaluation are complex, and it is not possible to model system operation in software only. Hardware/software emulation is required.

Even with hardware/software emulation, a simulation of a standard benchmark can take several days. Such simulations are inadequate for analyzing product lifetimes. Hence, sampling of system activity, in combination with hardware/software emulation is required to estimate system wearout for each sample, associated with specific variation in process parameters and wearout parameters.

CHAPTER 3

DEVICE-LEVEL WEAROUT MODELS

The first step in insuring reliable system operation is to bridge the gap between the established device level wearout models and system behavior at the architecture level. The current mean time to failure (MTTF) based high level reliability models, such as [74],[75], only provide us with crude, single point, reliability estimates based on the assumption that the system is a series failure system.

These methods assume an exponential failure rate distribution and that we can compute a MTTF for each mechanism and each block. However, each block is composed of a large number of elements, all failing at different rates, based on their temperature, electrical stress, and geometry.

Moreover, component failure rates are typically modeled with a Weibull or Lognormal distributions, rather than exponential distributions. Hence, the methodology to determine the MTTF for each block, as required in [74],[75], is not clear. Instead, we work with process-level models directly, and propagate these models to system-level models.

In this chapter we begin by presenting the detailed wearout models for microprocessor system components. Incorporating accurate electrical stress distributions for whole systems and functional units, accounting for the operating temperature and all vulnerable areas in layouts, our methodology establishes a link between the device level wearout models and the architecture level to estimate lifetimes more accurately for different wearout mechanisms.

Wearout mechanisms can be divided into two broad categories, the voltage (or E-field) dependent wearout mechanisms, such as NBTI, PBTI, GOBD, and BTDDB etc., and the current-stress dependent wearout mechanisms, such as EM and HCI. Due to the lack of higher level models for the progressive effect of these mechanisms, it is necessary to first model their effects at the device level and then abstract the models to the systeme level.

3.1 Backend Time-Dependent Dielectric Breakdown (BTDDB)

Dielectric breakdown is the irreversible local breakdown of a dielectric's insulation property. Time-dependent dielectric breakdown (TDDB) is dielectric breakdown that takes place after a constant application of an electric field (E), lower than the breakdown field, to the dielectric, as illustrated in Figure 3.1. TDDB results in the local development of a very small spot with increased conductivity compared to the rest of the dielectric, resulting in a change in the electrical characteristics of the dielectric [76]. In interconnects, TDDB of the low-k dielectric leads to catastrophic breakdown of the system.

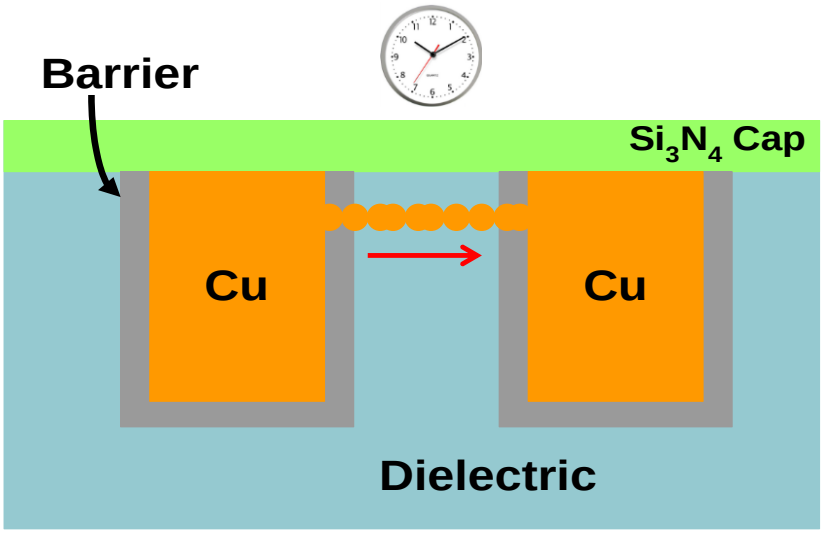


Figure 3.1: Cross section of an example dual-damascene Cu/Low-k interconnect under BTDDB.

The characteristic lifetime of a dielectric segment of the microprocessor, with vulnerable length, L_i, associated with linespace, S_i, is [61]-[63]

$$\eta = A_{BTDDB} L_i^{-1/\beta_i} exp(-\gamma E^m - E_a/kT), \quad (3.1)$$

where A_{BTDDB} is a constant that depends on the material properties of the dielectric, γ, is the field acceleration factor, m is one for the E model [77] and $1/2$ for the $\sqrt{E}$ model [78]. The electric field is a function of voltage, V, and the linespace, S, between the two lines surrounding a dielectric segment, i.e., $E=V/S$. The electric field, temperature (T), and geometry (L_i) determine the characteristic lifetime, η. The temperature dependence is modeled with the Arrhenius relationship [79], where k is the Boltzmann constant.

It should be noted that process data comes from test structures that are stressed with DC stress, while the microprocessor dielectrics undergo AC stress. Since BTDDB is most often (but not always) manifested as an abrupt and irreversible increase in the

leakage current when a dielectric segment is under constant bias stress at elevated temperature [80]-[83], the impact of switching on the dielectric segment is negligible. Hence, for segments of the microprocessor, it is sufficient to determine the time that each dielectric segment is under stress. To translate the DC stress of the test structure to the AC stress of the circuit, we compute the probability that each adjacent net has opposite voltages, α. To do this, we collect the electrical state profiles of each net within the microprocessor while running standard benchmarks [84] using FPGA emulation described in Section 4. First, we find the probability, p_i, that each net is at logic "1". We then compute the stress probability of a dielectric segment as the probability that the two adjacent nets are at different logic states. If the adjacent state probabilities are p_1 and p_2, then

$$\alpha = p_1(1 - p_2) + p_2(1 - p_1). \tag{3.2}$$

Equation (3.2) has been verified by comparing the exact stress durations of random-selected vulnerable dielectric segments from an example system layout with the ones calculated. The result, as illustrated in Figure 3.2, shows the percent errors are less than 15% for more than 80% of the selected samples. The high errors are mostly from the dielectric segments in deeper locations of the circuit. Since errors are accumulative, more activity propagation due to deeper stages leads to a bigger difference between real stress probabilities and calculated ones.

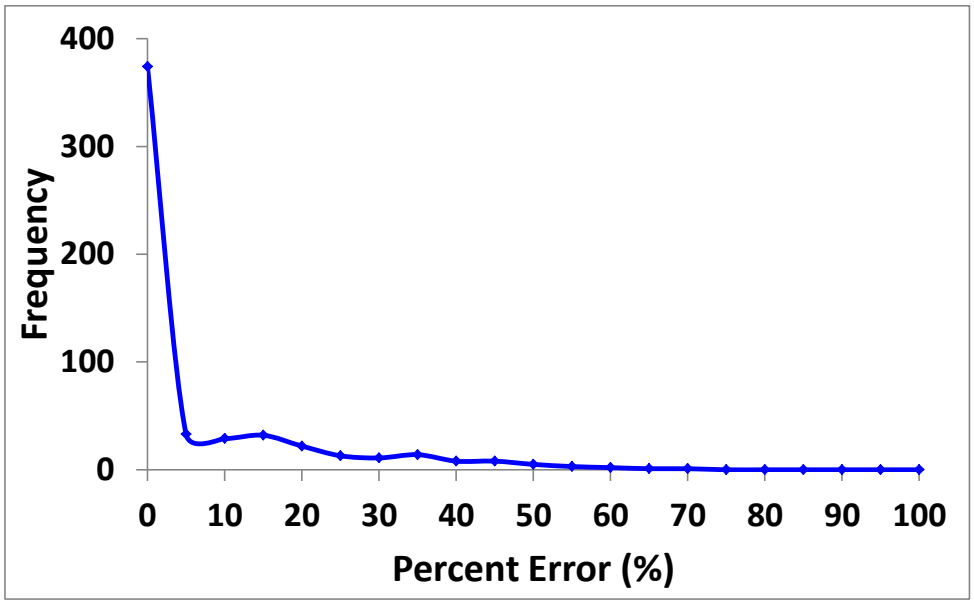


Figure 3.2: Percent error distribution of the random-selected dielectric segments.

3.1.1 Vulnerable Dielectric Area and Test Structures

In order to calculate the vulnerability of a layout to BTDDB, the BTDDB simulator operates by breaking down the dielectric in each layer and each block into dielectric segments. Each dielectric segment is characterized by a vulnerable length, L_i, and a linespace, S_i. The vulnerable length, L_i, is defined as the length of a block of dielectric between two copper lines separated by linespace S_i, illustrated in Figure 3.3. A given layout is analyzed by determining the pairs (S_i, L_i) for each layer for all linespaces.

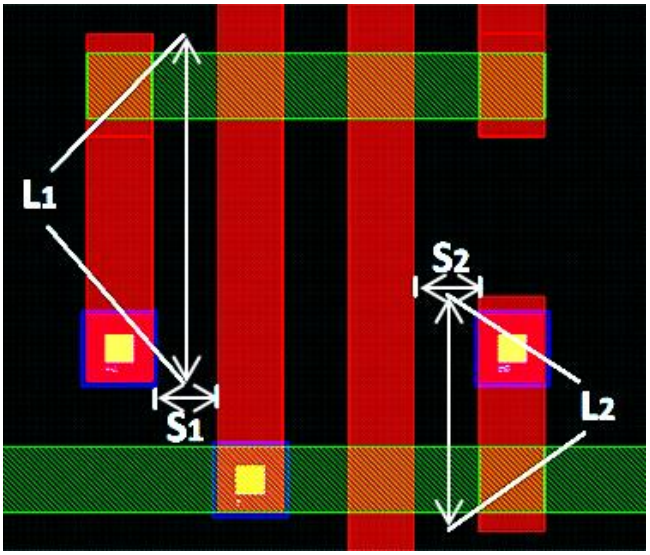


Figure 3.3: The vulnerable length associated with a linespace is shown. The rectangles are copper wires, surrounded by the backend dielectric.

Test structures have been designed to assess the impact of linespace and area on Cu/low-k TDDB. The details of the test structures, their design and results, are given in [9]. The test structure in Figure 3.4(a) is used to determine the lifetime of the dielectric between parallel tracks with a specific line spacing. This test structure has a fixed linespace, *S*, and vulnerable length, *L*. The vulnerable area is *LS*. To test the lifetime of such a feature, a voltage difference is applied between the two combs. The current between the combs is monitored to determine the time-to-failure. The data set from several samples is fit with a Weibull distribution to estimate η_t and β_t.

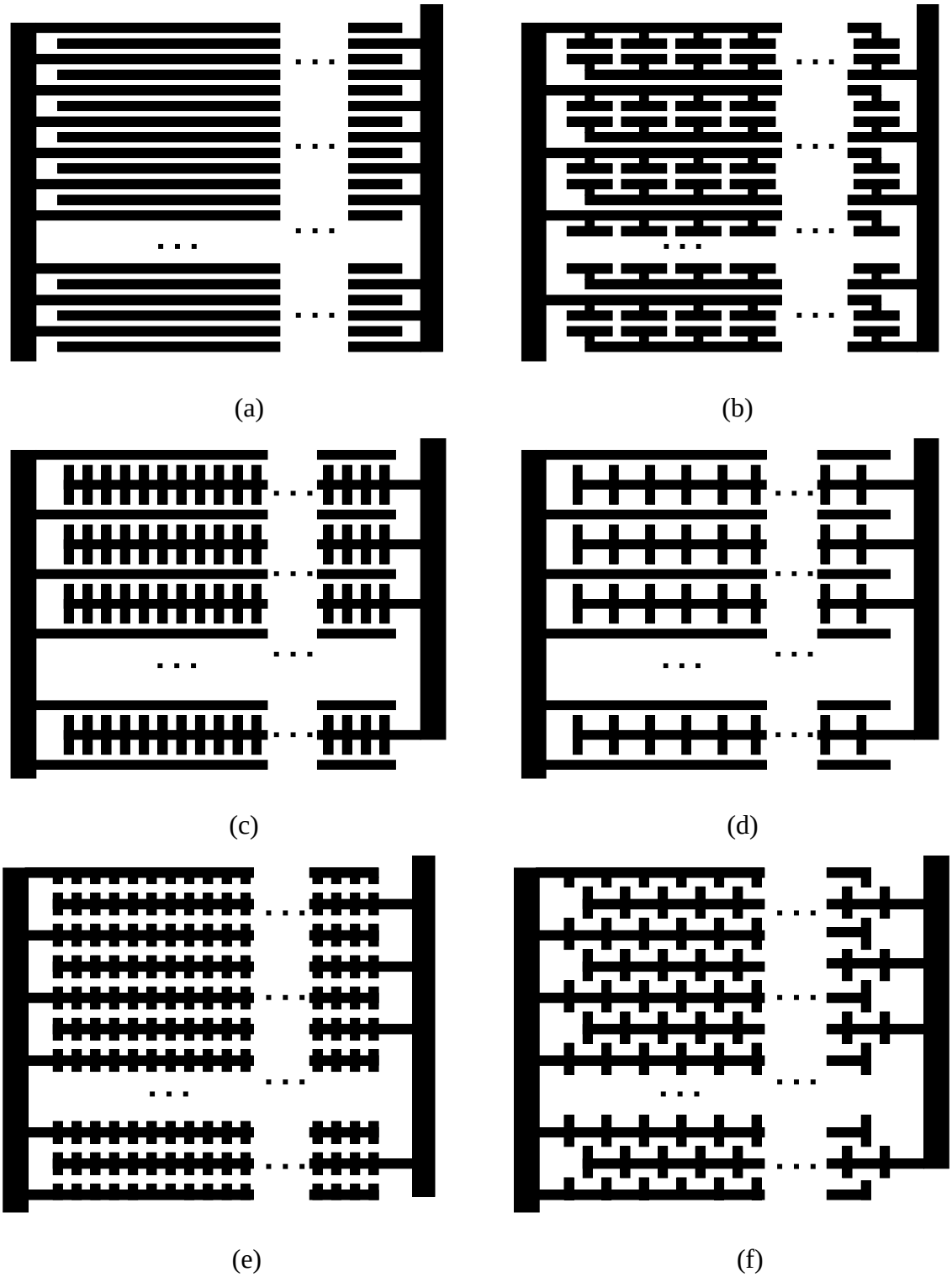


Figure 3.4: Top views of comb test structures to characterize the impact of geometry on time-dependent dielectric breakdown. (a) Standard comb structure, (b) PTT, (c) TLa, (d) TLb, (e) TTa, and (f) TTb.

Because the features on a chip differ from a test structure layout, area scaling must be performed to adjust the lifetime to take into account the difference in vulnerable area between the chip and the test structure. To do this, let L_t and L_i be vulnerable lengths of the test structure and chip, i.e. the length of the lines that run in parallel in the test structure and chip, respectively, with the same linespace, *S*. η_t is determined by stressing a test structure with linespace *S* and vulnerable length L_t. Then the corresponding characteristic lifetime for that feature in the chip is

$$\eta_i = \eta_t \left(\frac{L_t}{L_i}\right)^{1/\beta}. \tag{3.3}$$

Test structures that have several irregular features have been designed in order to determine any impact of field enhancement. Figure 3.4(b)-(f) shows the top views of these test structures and the fragments of these test structures are shown in Figure 3.5. PTT emphasizes the electric field between parallel routing tracks that end at the same point. TLa and TLb emphasize the electric field between line ends and perpendicular lines. TLb includes additional fringing fields, since the line ends are more widely spaced. TTa and TTb emphasize electric fields between line ends. In TTa, the line ends abut, and in TTb the line ends are in parallel tracks. TLa, TLb, TTa, and TTb have 528 line ends each. The separation between line ends is the same for all test structures.

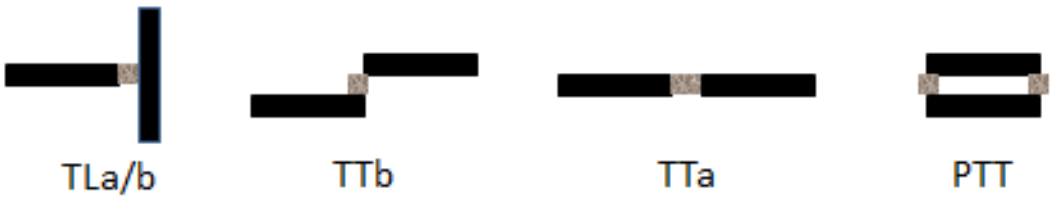


Figure 3.5: Vulnerable line ends that need to be extracted from a layout.

All test structures in Figure 3.3 have the same minimum line space, 140nm. If the drawn line space is consistent with the printed line space, then the relative influence of each geometry would be the same. Moreover, the number of vulnerable line ends for each geometry in Figure 3.4(c)-(f) is constant, i.e. 528 line ends each. Hence, when comparing the test structure in Figure 3.4(c)-(f), no area scaling is required (using equation (3.3)) when comparing the results. On the other hand, we require area scaling to determine if the test structures in Figure 3.4(b)-(f) result in an increase failure rate in comparison with parallel lines, as in Figure 3.4(a). The test structures were tested at 3.6MV/cm and at 150˚C, and the current between the lines was monitored . A current limit of 10 μA was set to detect dielectric breakdown.

To account for irregular features, the counts of the features are extracted from the layout. Each add additional parameters, η_{PTT}, β_{PTT}, $\eta_{TLa/b}$, $\beta_{TLa/b}$, η_{TTa}, β_{TTa}, η_{TTb}, and β_{TTb} to (2.2) and (2.3). These parameters depend on the number of minimally spaced line ends in each category of the layout. Let's consider the computation of $\eta_{TLa/b}$ for the sake of illustration. Let's suppose the test structure has N_{test} minimally spaced line ends, from which η_{test} and $\beta_{TLa/b}$ are computed. Then, for a layout with N_{chip} similar line ends, by area scaling

$$\eta_{TLa/b} = \eta_{test}\left(\frac{N_{test}}{N_{chip}}\right)^{1/\beta_{TLa/b}}. \tag{3.4}$$

3.1.2 Test Results

Let's suppose that there is no field enhancement due to any of the features in Figure 3.4(b)-(f). Then the lifetime data from the test structure in Figure 3.4(a) would be sufficient to predict the lifetimes of the test structures in Figure 3.4(b)-(f). We make this

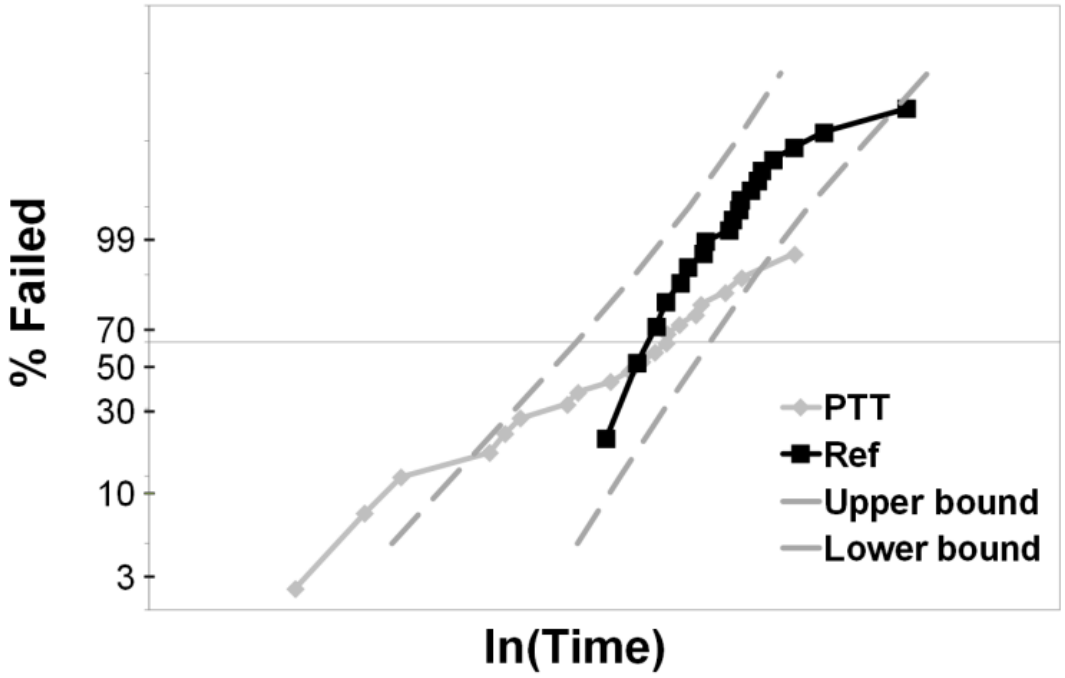
assumption and extract the vulnerable length, *L*, and linespace, *S*, for the test structures in Figure 3.4(a)-(f).

Next, we compare the measured Weibull curves for the test structures in Figure 3.4(b)-(f) with the Weibull curve from the standard comb test structure with the same linespace, *S*, in Figure 3.4(a), area scaled [36] – by using the Poisson area scaling invariance of the Weibull distribution – to match the vulnerable length of the test structures in Figure 3.4(b)-(f).

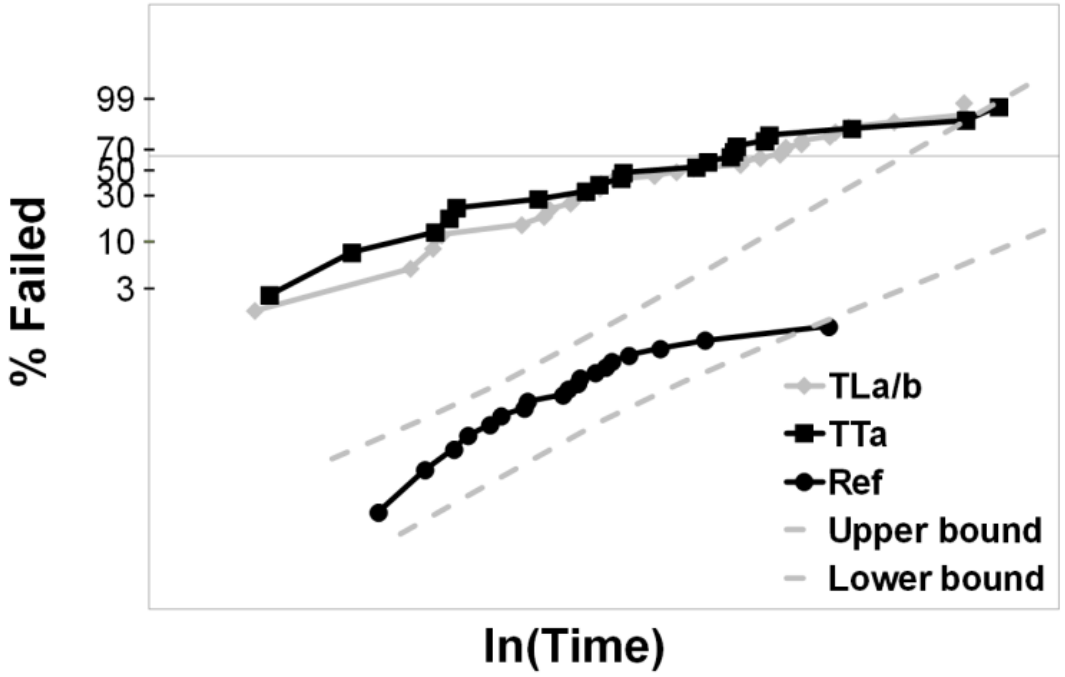
Specifically, let $N = L_t/L_i$ be the ratio of vulnerable length, where L_t corresponds to the vulnerable length of the standard comb structure in Figure 3.4(a) and L_i corresponds to the vulnerable length in one of Figure 3.4(b)-(f). To area-scale the standard comb structure to give us the lifetime distribution for a different (smaller) vulnerable area, we plot

$$\ln \eta_t = \ln TF - \frac{1}{\beta} ln\left(-\frac{1}{N} ln(1 - P(TF))\right) . \quad (3.5)$$

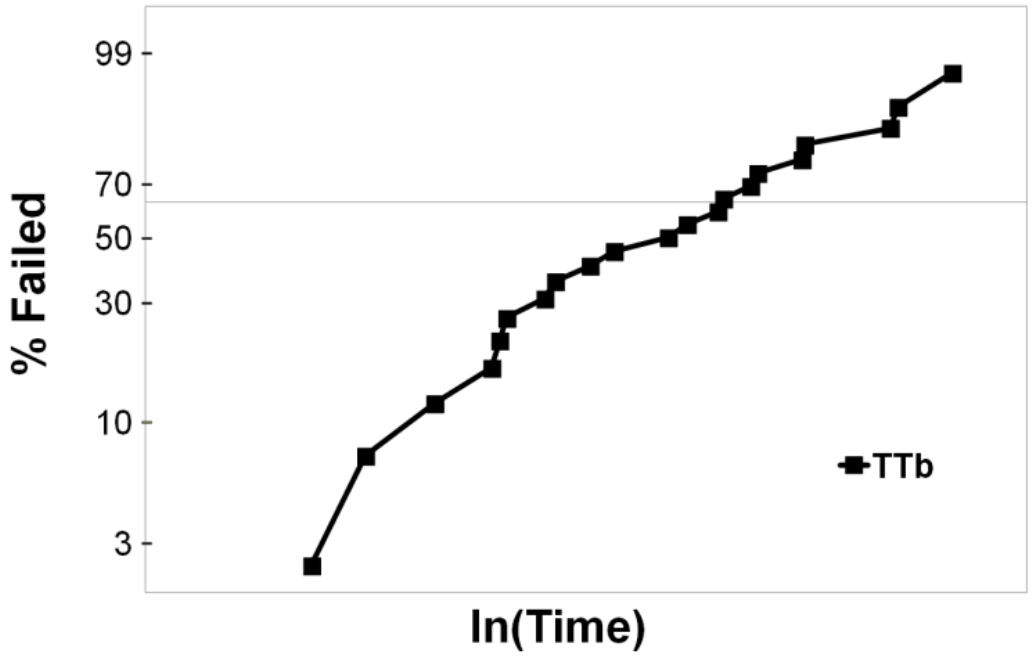
The line end features in Figure 3.5 were found to have a significant impact on lifetime. The data collected from the test structures is presented in Figure 3.6. An area scaled version of a standard comb test structure is included for comparison. It can be seen that all test structures (PTT, TLa, TLb, TTa, and TTb) result in a significantly reduced lifetime in comparison with the reference test structure. The data also indicate that TLa and TLb fail at the same rate, showing that fringing fields are not significant. The data from these two test structures can be merged to determine a single model. TTa has an improved lifetime, in comparison with TLa/b. No reference curve is included for comparison for TTb because TTb has no vulnerable length.


% Failed
99
70
50
30
10
3
PTT
Ref
Upper bound
Lower bound
ln(Time)

(a)


% Failed
99
70
50
30
10
3
TLa/b
TTa
Ref
Upper bound
Lower bound
ln(Time)

(b)



(c)

Figure 3.6: Data collected from (a) PTT vs. the reference structure, (b) TLa, TLb, and TTa vs. the reference structure, and (c) TTb. 2σ confidence bounds are included for the area scaled reference test structure.

Since the test results indicate all of the line ends create an increased vulnerability for PTT, TLa, TLb, TTa, and TTb and fail more rapidly, the counts of the vulnerable line ends with these geometries need to be incorporated separately from the vulnerable length in the simulator when estimating the wear-out of a full chip.

3.1.3 Model Constructions for Irregular Geometries

A model was extracted for PTT, TLa/b, TTa, and TTb. The model for TTa and TTb was found with the standard method, involving fitting a linear function to the data to find η_{TTa}, β_{TTa}, η_{TTb}, and β_{TTb}.

Extraction of the model for TLa/b and PTT is more complex, since these structures combine both line ends and vulnerable length. Figure 3.7 shows one TTa/b line end and two PTT line ends, together with the vulnerable length extracted. If one finds $\eta_{TLa/b}, \beta_{TLa/b}$, η_{PTT}, and β_{PTT} by fitting a linear function, then the model would include both the impact of the line ends and the vulnerable length. For circuit analysis purposes, it is necessary to eliminate the effect of vulnerable length to create a model for line ends only. To find the model for line ends, it is necessary to subtract the effect of vulnerable length. Let η_{TS} and β_{TS} be the measured data from the test structures TLa/b and PTT. For each of these test structures we need to determine η_{ends} and β_{ends}, after eliminating the component due to vulnerable area, η_{area} and β_{area}. The parameters, η_{area} and β_{area}, are determined from the area scaled data from the reference test structure.

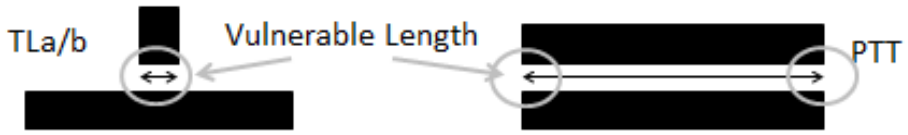


Figure 3.7: Vulnerable length and line ends extracted from test structure TLa/b and PTT. The vulnerable length is indicated with arrows and the line ends are indicated with circles.

Relying on equations (2.2) and (2.3), we have that

$$1 = \left(\frac{\eta_{TS}}{\eta_{ends}}\right)^{\beta_{ends}} + \left(\frac{\eta_{TS}}{\eta_{area}}\right)^{\beta_{area}} \tag{3.6}$$

and

$$\beta_{TS} = \beta_{ends}\left(\frac{\eta_{TS}}{\eta_{ends}}\right)^{\beta_{ends}} + \beta_{area}\left(\frac{\eta_{TS}}{\eta_{area}}\right)^{\beta_{area}}. \tag{3.7}$$

Rearranging the equations results in

$$\beta_{ends} = \frac{\beta_{TS} - \beta_{area}\left(\frac{\eta_{TS}}{\eta_{area}}\right)^{\beta_{area}}}{1-\left(\frac{\eta_{TS}}{\eta_{area}}\right)^{\beta_{area}}} \tag{3.8}$$

and

$$\eta_{ends} = \eta_{TS}\left(1 - \left(\frac{\eta_{TS}}{\eta_{area}}\right)^{\beta_{area}}\right)^{-1/\beta_{ends}}. \tag{3.9}$$

These equations were used to extract the model for TLa/b. Because of the large separation between the data and the reference, the shift in η and β due to subtracting the impact of vulnerable area is less than 0.1% and 1%, respectively. This is illustrated in Figure 3.8.

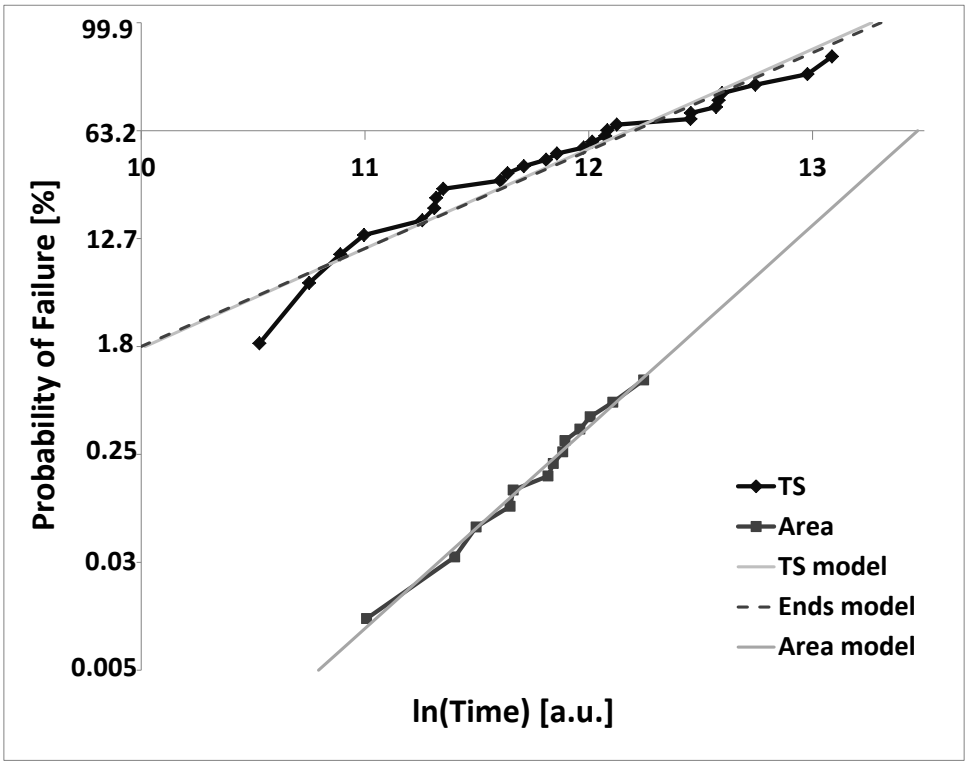


Figure 3.8: Data collected from TLa/b vs. the reference structure. The models for the data from the test structure and the line ends, after subtracting the effect of area are nearly indistinguishable.

Equations (3.8) and (3.9) cannot be used for PTT. This is because equations (2.3) and (3.7) were derived by finding the probability density function of the combined failure rate as a function of the underlying parameters, converting to the Weibull probability scale (i.e. $\ln(-\ln(1-P))$), and evaluating the slope at the characteristic lifetime, η. As can be seen from Figure 3.5(a), PTT impacts lower probabilities, and the slope is not well defined at the x-intercept of the Weibull plot.

Instead, we need to find η_{ends} and β_{ends} by defining the probability density function for the test structure as a function of *TF*, for any value of *TF*, i.e.,

$$P(TF) = 1 - exp\left(-\left(\frac{TF}{\eta_{TS}}\right)^{\beta_{TS}}\right). \tag{3.10}$$

Since this probability density function results from two independent mechanisms, we also have that

$$P(TF) = 1 - exp\left(-\left(\frac{TF}{\eta_{ends}}\right)^{\beta_{ends}} - \left(\frac{TF}{\eta_{area}}\right)^{\beta_{area}}\right). \tag{3.11}$$

Hence,

$$\left(\frac{TF}{\eta_{ends}}\right)^{\beta_{ends}} + \left(\frac{TF}{\eta_{area}}\right)^{\beta_{area}} = \left(\frac{TF}{\eta_{TS}}\right)^{\beta_{TS}}. \tag{3.12}$$

We solve for the unknowns, η_{ends} and β_{ends}, by finding the best fit to the data in the range where end failures are dominant, through linear regression. The results are shown in Figure 3.9.

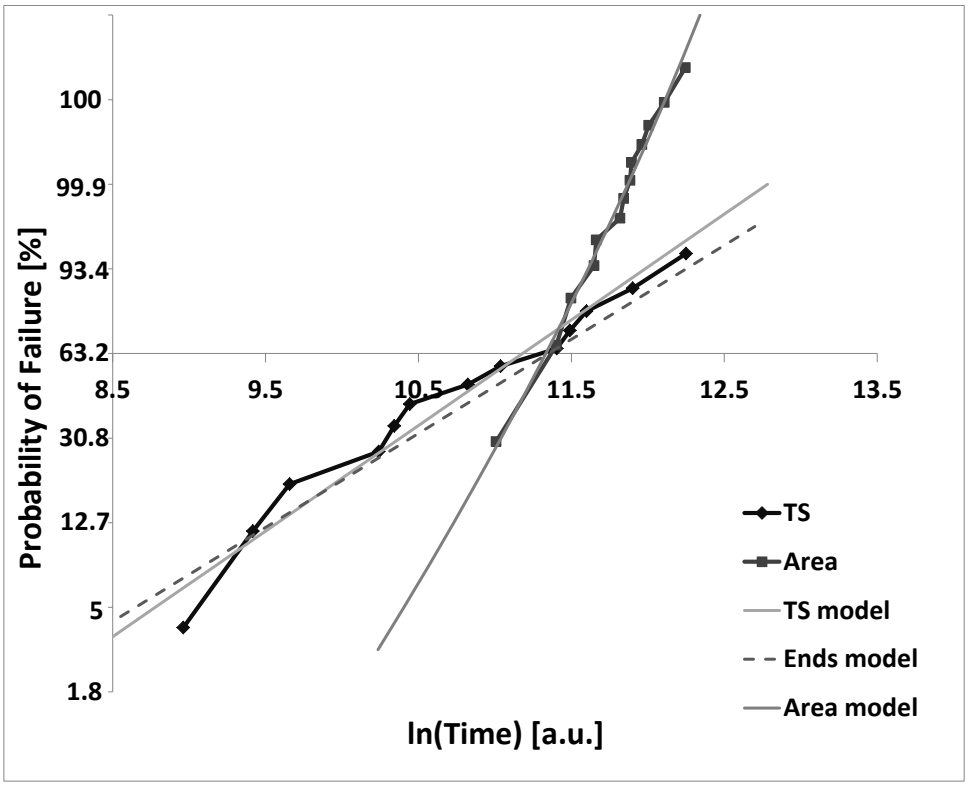


Figure 3.9: Data collected from PTT vs. the reference structure. The graph shows the models for the data from the test structure vs. the line ends, after subtracting the effect of area.

3.1.4 Vulnerable Length and Feature Extraction

BTDDB requires the determination of the vulnerable length of the dielectric segments as a function of linespace. A layout extraction tool has been developed using the standard object oriented programming languages. A detailed description of the algorithm is given in Algorithm 3.1.

Input: The maximum line spacing S_{max} and a layout L
Output: Tables of vulnerable lengths (VulnerableLengthTable) and new features (TLab, TTa, TTb, PTT)

```
for each metal layer m do
   LineDataX (m) ← ReadLines (L);  // BucketSort
   LineDataY (m) ← ReadLines (L);  // BucketSort
   TTa (m) ← 0;  TTb (m) ← 0;  PTT (m) ← 0;  TLab (m) ← 0;
   c ← 1;
   n ← 2;
   while c<N_line do  // N_line: # lines in LineDataY
      L1 ← LineDataY (m,c);  // c-th line
      L2 ← LineDataY (m,n);  // n-th line
      if  Spacing (L1, L2)<=S_max then
         TLab (m) += CheckTLab (L1, L2);  // check TLab between L1 and L2
         TTa (m) += CheckTTa (L1, L2);  // check TTa between L1 and L2
      end
      n ← Adjust (c, n);
      L2 ← LineDataX (m,n);
      if  Spacing (L1, L2)<=S_max then
         TLab (m) += CheckTLab (L1, L2);  // check TLab between L1 and L2
      end
      n ← Adjust (c, n);
      L2 ← LineDataY (m,n);
      if  Spacing (L1, L2)<=S_max then
         PTT (m) += CheckPTT (L1, L2);  // check PTT between L1 and L2
         TTb (m) += CheckTTb (L1, L2);  // check TTb between L1 and L2
         VulnerableLengthTable (m) ← VulnerableLength (L1, L2);
         LineDataY (m) ←Split (L1, L2);
         n ← Adjust (c, n);
      end
      n ← Adjust (c, n);
   end
end.
```

Algorithm 3.1: Layout extraction flow

Vulnerable area and features are extracted by comparing two lines in a given layout. Since tens of millions of lines exist in each metal layer in a layout, it is necessary to find two adjacent lines forming a vulnerable area or a feature in a short time. Therefore, vulnerable area and features are extracted as follows. First of all, lines are read from a

given layout, sorted by the bucket sort algorithm, and stored in two separate data variables, LineDataX and LineDataY. The lines in LineDataX (or LineDataY) are sorted in the ascending order of the x-coordinates (or y-coordinates) of the bottom left corner of the lines. If two lines have the same x-coordinate (or y-coordinate), they are sorted in the ascending order of the y-coordinates (or x-coordinates) of the bottom left corner of the lines. Then, the extraction process starts by comparing the first (L_1) and the second (L_2) lines in the first bucket of LineDataY. Since each metal layer has a preferred routing direction (horizontal or vertical), the preferred routing direction is assumed to be horizontal in this explanation. Then, the y-coordinates of the two lines in the same bucket are the same, so they can form TTa or TLa/b depending on the distance between them and the direction (horizontal or vertical) of the lines. Whether or not they form TTa (or TLa/b) or not, the first line does not form any features with other lines in the same bucket because the second line lies between the first and the other lines in the bucket. Then the index of the second line is adjusted to find TLa/b between L_1 and other vertical lines. If L_1 is horizontal, L_2 should be vertical to form TLa/b with L_1, so LineDataX is searched based on the x-coordinate of the bottom right corner of L_1 to find L_2 that can form TLa/b with L_1.

TTb or PTT is extracted by comparing two lines in different buckets (lines in different buckets have different y-coordinates) in LineDataY. Therefore, the index of L_2 is adjusted and whether they form TTb or PTT is checked. If TTb or PTT is found, the flag of the edge of L_1 forming TTb or PTT with L_2 is set. By setting the flag, counting TTb or PTT formed by L_1 and L_3 is avoided when the x-coordinate of L_3 is the same as that of L_2 and the distance between L_1 and L_3 is less than the maximum line spacing.

After extracting irregular features formed by L_1 and its adjacent lines, the vulnerable length between L_1 and L_2 is extracted. If the vertical spacing is less than or equal to the maximum line spacing, a vulnerable area surrounded by these two lines exists, so the vulnerable length is added to the vulnerable length table. Then, L_1 is split into one or two new lines, they are inserted into LineDataY, and L_1 is removed from LineDataY.

Figure 3.10 shows an example with four lines, S_1, S_2, S_3, and S_4. The algorithm starts with the first line segment, S_1. S_1 forms PTT with S_2 (when the distance between them is smaller than the maximum line spacing) as shown in Figure 3.10(b). They also form a vulnerable area as shown in Figure 3.10(c), so the vulnerable length is added to the vulnerable line table. Then, S_1 is split into new lines. In this example, only one new line (S_{1-1}) is created because the left boundaries of S_1 and S_2 are aligned as shown in Figure 3.10(d). After inserting S_{1-1} into LineDataY, L_1 is set to S_{1-1} and L_2 is set to S_2, and the extraction process is repeated between them. TTb exists between S_{1-1} and S_2 as shown in Figure 3.10(e). Similarly, TTb exists between S_{1-1} and S_3 in Figure 3.10(f). Since S_{1-1} does not overlap with other lines, L_1 is set to S_2 and L_2 is set to S_4 by the index adjustment function. In the next extraction process, TLa/b is extracted between S_2 and S_4 in Figure 3.10(g), and TTa is extracted between S_2 and S_3 in Figure 3.10(h).

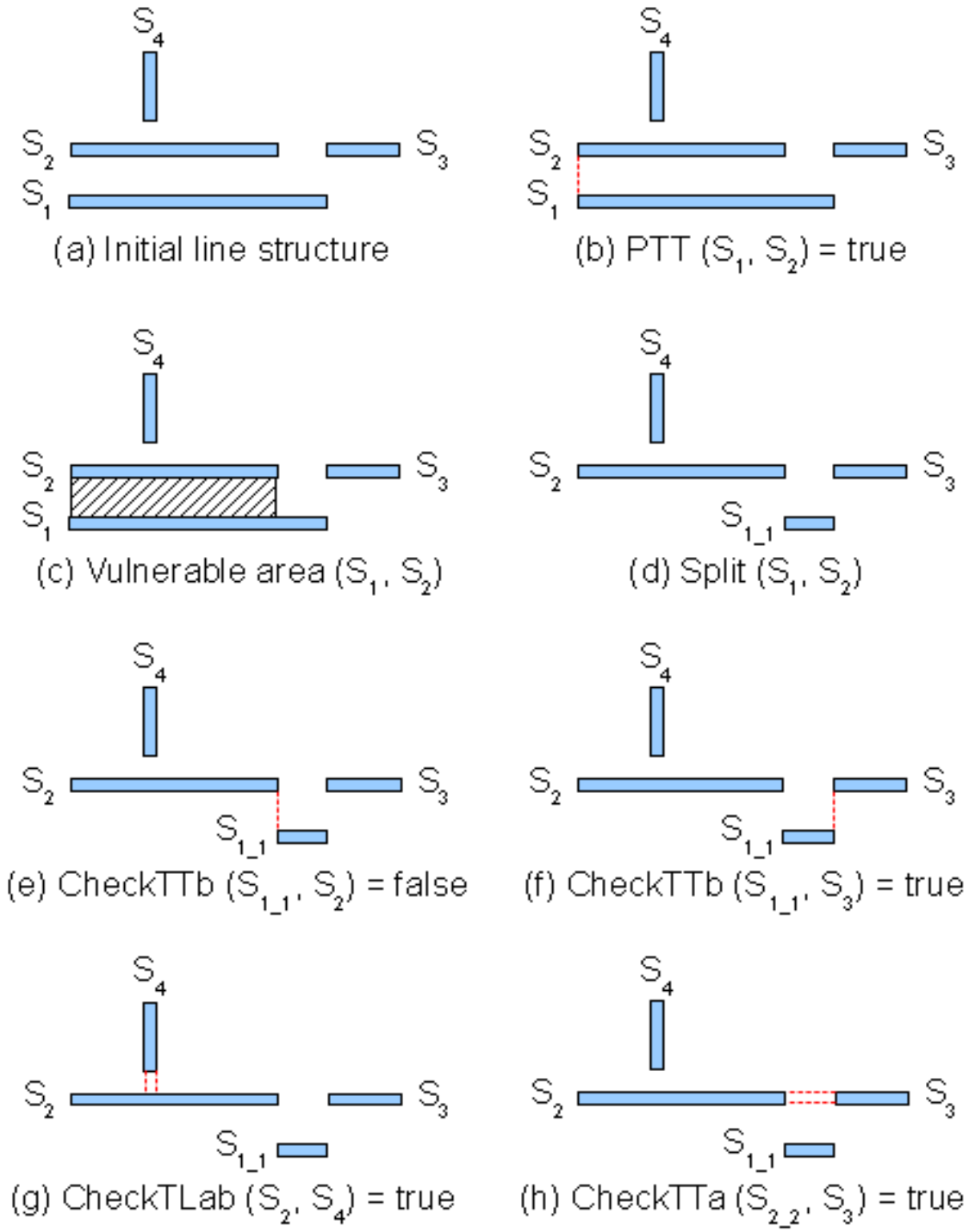


Figure 3.10: (a) Initial line structure. (b) PTT is extracted from S1 and S2. (c) Vulnerable length between S1 and S2 is extracted. (d) Postpocessing after vulnerable area extraction. (e) TTb does not exist between S1_1 and S2. (f) TTb is extracted from S1_1 and S3. (g) TLa/b is extracted from S2 and S4. (h) TTa is extracted from S2 and S3.

Complexity of vulnerable feature extraction is O(n), where n is the number of features, since bucket-sort algorithm is used. Complexity of extracting statistics from features is also O(n), because the bucket is scanned from the bottom most element, and the maximum number of features within a fixed distance from an element is constant. Hence, layout feature extraction is linear in terms of the number of geometries analyzed and is linear as a function of the area of a chip.

After extraction of the dielectric segments' length and linespace, each dielectric segment is linked to its thermal and stress profile in order to compute its characteristic lifetime with equation (3.1). Temperature is a function of the location of the segment in the layout, and stress is a function of the state probabilities of the adjacent nets.

3.2 Electromigration (EM)

EM refers to the dislocation of metal atoms caused by momentum imparted by electrical current in interconnects and vias. The vulnerable location is the interconnect/via interface, where a void can form, as illustrated in Figure 3.11. Specifically, vias are damaged by downstream electron flow, from the via to the metal below it. This is because the via and the line below it are formed by separate deposition steps, which creates a vulnerable interface. Hence, although EM can be observed in interconnect lines, it is much more likely to be seen at via interfaces [85],[86]. Therefore, this work focuses on EM in vias, rather than in the significantly less vulnerable interconnect lines. The characteristic lifetime, η , of a via due to EM can be modeled as [87]-[90]:

$$\eta = A_{EM}\, T/j, \tag{3.13}$$

where T is temperature, j is the current density, and A_{EM} is a technology dependent constant that takes into account the velocity of the void, the resistivity of the metal, surface diffusivity, surface thickness, the thickness of the line, and the via size. The data on EM used in this study comes from Choi's experimental data [87].

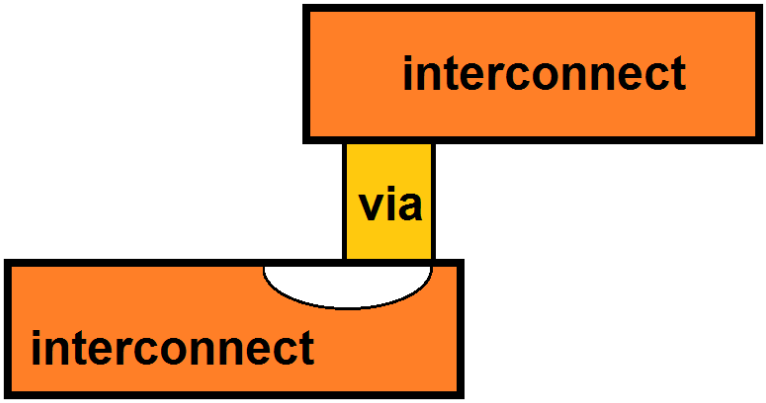


Figure 3.11: An example vulnerable interconnect/via interface under EM.

In order to calculate the vulnerability of a layout to EM, the EM simulator operates by determining the characteristic lifetime of each via within each interconnect segment in the microprocessor layout. To do this, we find the current density of each interconnect, by collecting the switching activity profiles of each interconnect segment while running standard benchmarks [84] using FPGA emulation. When calculating the corresponding current density for each via on an interconnect, since the current always flows from a via on one end of an interconnect to the vias on the other end, we assume that one of the vias on an interconnect segment experiences EM degradation during the rising/falling transitions and the rest of the vias experience degradation during the opposite transition for signal nets. On the other hand, only one of the vias in each power

supply/ground net experiences degradation, because current flow in power supply/ground nets is unidirectional.

The Automatic Place and Route (APR) tool [91] has been used to collect the via locations and total number of vias connected to each interconnect segment, v_i, when the system layouts are generated. The computational cost is O(1). One via is assumed to be impacted by rising/falling transitions and the rest, (v_i - 1) are assumed to be impacted by the opposite transition. The corresponding current density, $j_{interconnect}$, for rising or falling transitions, is averaged over each via at each end of an interconnect, to give us the average via current densities, $j_{via} = j_{interconnect}$ and $j_{via} = j_{interconnect}/(v_i - 1)$, respectively. To verify average via current densities, the actual current densities of randomly-selected vias are calculated based on the real interconnect geometries and compared with their average via current densities. The result, as illustrated in Figure 3.12, shows the percent errors are less than 10% for more than 80% of the selected samples.

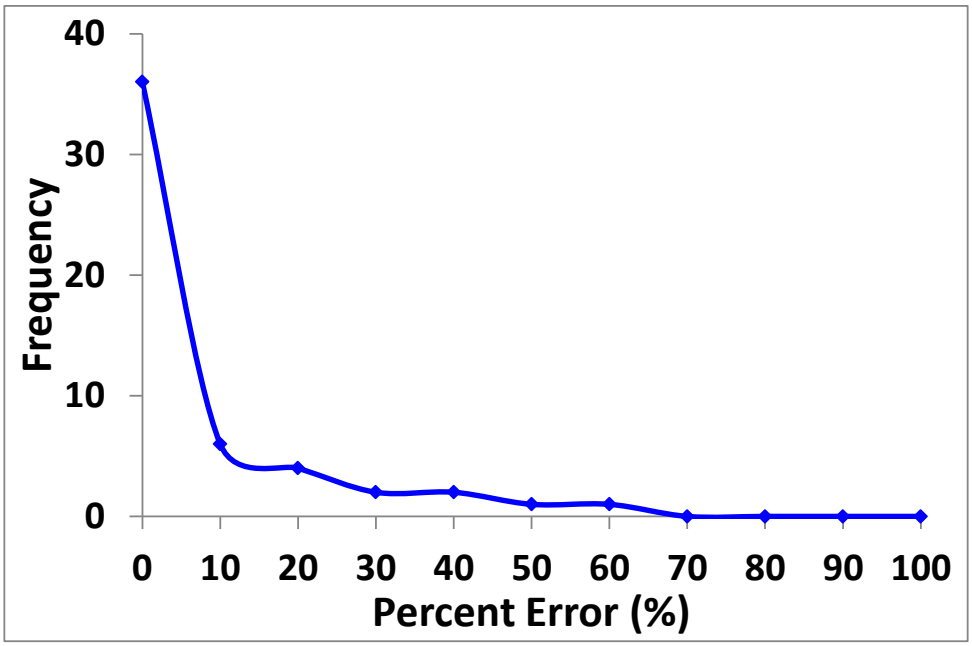


Figure 3.12: Percent error distribution of the random-selected via current densities.

The location of each via/interconnect segment is determined to provide a link to its thermal profile, to find the characteristic lifetime of each via with equation (3.13).

3.3 Stress-Induced Voiding (SIV)

SIV damage is caused by the directionally biased motion of atoms in interconnects due to mechanical stress caused by thermal mismatch between metal and dielectric materials, as illustrated in Figure 3.13. As with EM, the failure site is at the via interfaces. This interface is vulnerable because the via and the line below it are formed by separate deposition steps. SIV depends on the geometry above a via, because larger geometries result in more material expansion and contraction with temperature, which in turn creates greater stress at the vulnerable via interface. Based on the SIV dependence

on both temperature and geometric linewidth of the interconnect above a via, the characteristic lifetime, η, of a via under SIV is given by [92],[93]:

$$\eta = A_{SIV} W^{-M} (T_0 - T)^{-N} exp(E_a / kT) \tag{3.14}$$

where W is the linewidth, M is the geometry stress component, T_0 is the stress-free temperature, N is the thermal stress component, and A_{SIV} is a constant. SIV depends on switching activity to the extent that switching activity increases temperature. The data used in our study of SIV comes from Yao's experimental data [92].

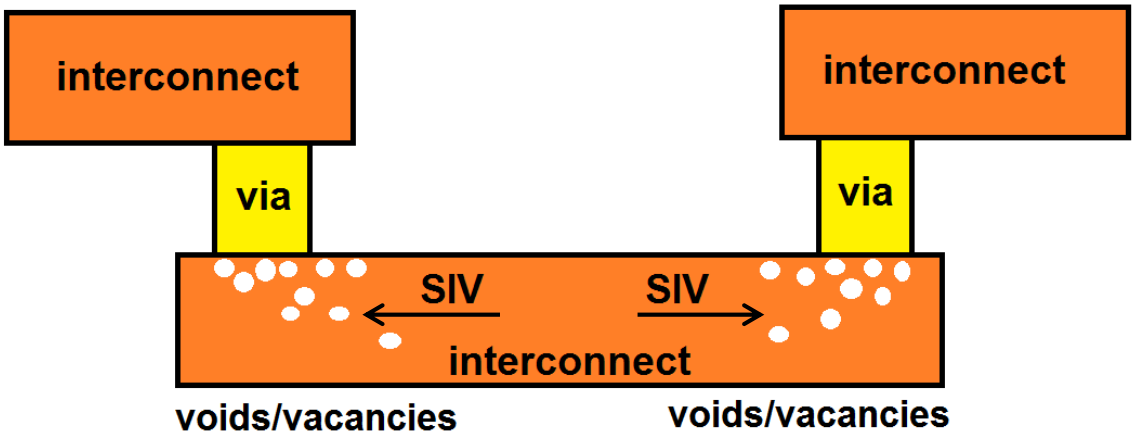


Figure 3.13: An example vulnerable interconnect/via interface under SIV.

In order to find the lifetime of each via with equation (3.14), the width of the interconnect segment above each via is extracted from the layout, and the location of each via is determined, to provide a link to the thermal profile.

3.4 Negative/Positive Bias Temperature Instability (NBTI/PBTI)

Bias temperature instability, as the name suggests, causes instability in device behavior and is a result of the bias stress applied to it. NBTI is the degradation of a PMOS device under negative gate stress, and PBTI is the degradation of an NMOS device under positive gate stress. NBTI and PBTI result in shifts in device parameters, such as threshold voltage, transconductance, device mobility, etc., but are generally identified by shifts in the threshold voltage.

Historically, BTI was only a major concern for PMOS devices, with NMOS devices showing comparatively negligible degradation. However with the introduction of high-k metal gate stacks for sub-45 nm technology nodes, degradation in NMOS devices due to positive bias has increased, with large degradation observed for both types of devices [94].

The threshold voltage drift caused by BTI is a function of stress time and recovery (non-stress) time, as illustrated in Figure 3.14, and can be modeled. A model of threshold voltage and its shift as a function of stress has three components. There is a model of the initial distribution, a model of the mean shift as a function of time under stress and recovery, and a model of the standard deviation of the shift, modeling the random variation of the change in threshold voltage for devices that experience identical stress profiles.

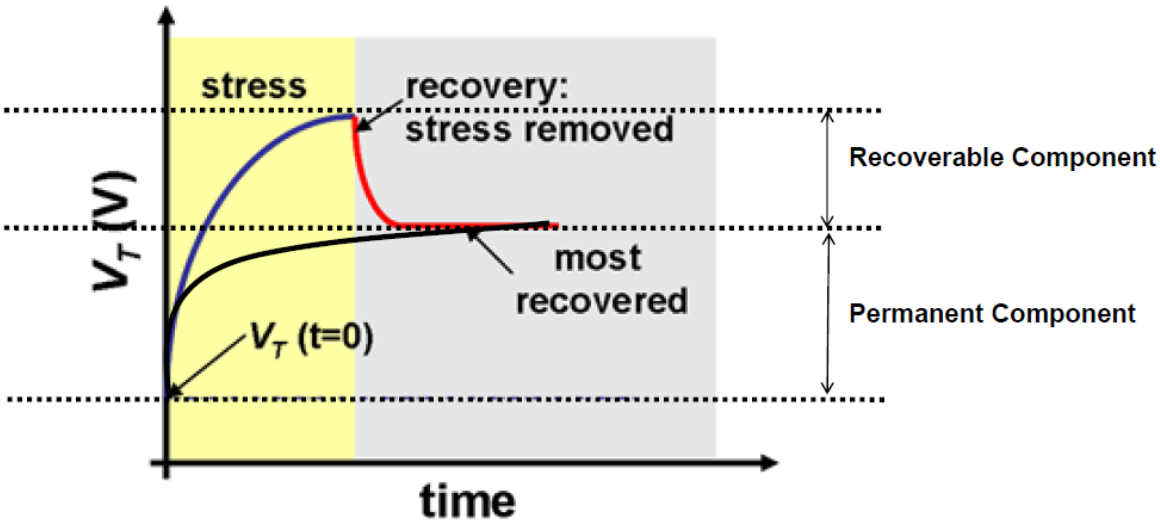


Figure 3.14: The threshold voltage drift caused by BTI is a function of stress time and recovery (non-stress) time.

The initial distribution is generally assumed to be Normal.

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 [20],[95]-[97]:

$$\Delta V_{tp/tn}(DC) = \phi(T, E_F)(A + \mathrm{Bln}(t_{DC})), \tag{3.15}$$

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:

$$\phi(T, E_F) = \phi_0 g(E_F) e^{-E_a/kT}, \tag{3.16}$$

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 [98]. 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 is a nonlinear function modeled as (α) , where $g(1) = 1$ and $g(0)$=0 [95]. 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,

$$\Delta V_{tp/tn} = \phi_0 e^{-E_a/kT} g(t_{stress}/(t_{stress} + t_{rec})) \cdot (A + Bln(t_{stress} + t_{rec})) \quad (3.17)$$

where ϕ_0 is a constant. The constants were obtained from the experimental results in [99].

Finally, there is a random component, i.e. $\sigma(\Delta V_{tp/tn})$. This is an exponential function, as noted in [20]:

$$\sigma(\Delta V_{tp/tn}) = e^{-\lambda} \Delta V_{tp/tn} \quad (3.18)$$

where λ is a constant. Hence, as time progresses for the i^{th} device:

$$(\Delta V_{tp/tn})(i) = \Delta V_{tp/tn} + \xi_i \sigma(\Delta V_{tp/tn}) \quad (3.19)$$

where ξ_i is a random number generated from a standard Normal distribution.

The degradation of threshold voltage results in longer delays at the circuit level, which eventually results in failure of circuit performances. For any circuit component, a threshold can be determined, such that shifts in the threshold voltage results in circuit-level failure, as was demonstrated in [100].

3.5 Hot Carrier Injection (HCI)

HCI describes the phenomenon by which carriers at a MOSFET's drain gain sufficient energy to be injected into the gate oxide and cause degradation of some device parameters. This occurs as carriers shoot out from the source of a MOSFET, accelerate in the channel, and experience impact ionization near the drain end of the device, as

illustrated in Figure 3.15. The damage can occur at the interface, within the oxide and/or within the sidewall spacer. Interface-state generation and charge trapping induced by this mechanism result in degradation of some MOSFET parameters, such as threshold voltage, transconductance, channel mobility and drain saturation current.

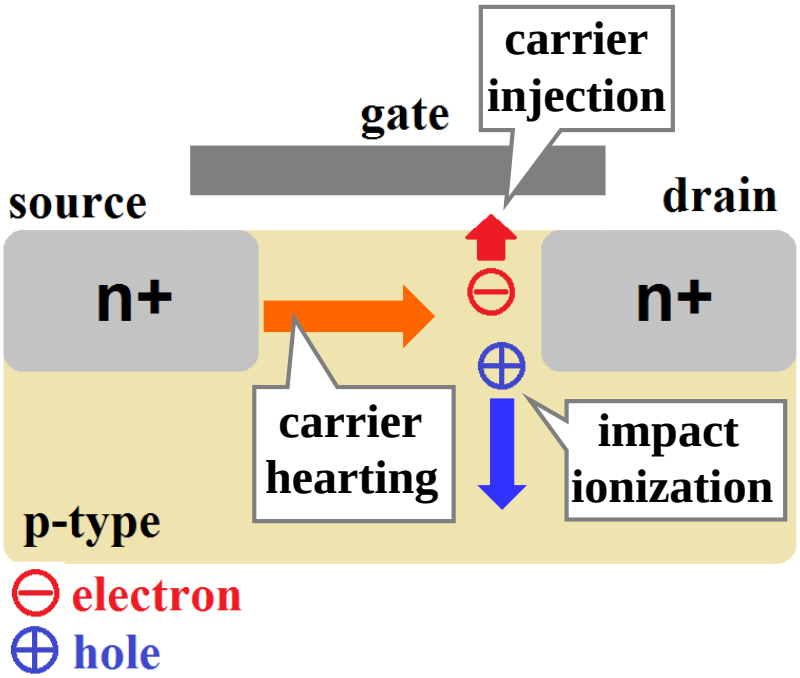


Figure 3.15: Carriers shoot out from the source of a NMOS, accelerate in the channel, and experience impact ionization near the drain end of the device.

Historically, HCI was only a major concern for nMOS devices, with pMOS devices showing comparatively negligible degradation because (a) holes have a smaller impact ionization rate and (b) holes face a higher $Si - SiO_2$ barrier than electrons. However, subsequent reports have revealed that HCI effects on pMOS devices has also been observed [101]. The shifts in threshold voltage and transconductance are proportional to the average trap density, which in turn is inversely proportional to the effective channel length. In addition, 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 can be modeled as [102]:

$$\Delta V_{tp/tn} = A_{HCI}(r_{trans} t_{stress} t_{trans})^n \tag{3.20}$$

where r_{trans} is the frequency-dependent transition rate, t_{stress} is the stress time, t_{trans} is the transition time, and A_{HCI} is a technology dependent constant that depends on the inversion charge, the trap generation energy, the hot electron mean free path, and other process-dependent factors. The data used in our study of HCI comes from the experimental data in [103],[104].

From the perspective of circuit operation, HCI and BTI stress have different time windows. HCI stresses devices only during the dynamic switching period when current flows through the device, whereas BTI stresses devices as a function of logic state. The stress time windows of NBTI, PBTI and HCI for an inverter circuit are illustrated in Figure 3.16 as an example.

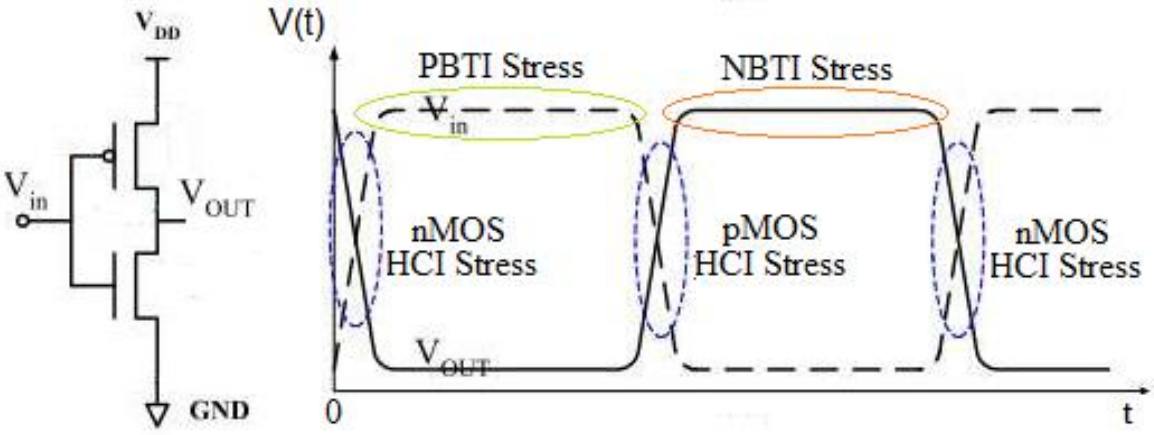


Figure 3.16: Stress-time windows of NBTI, PBTI and HCI for an inverter.

3.6 Gate Oxide Breakdown (GOBD)

GOBD is one of the key reliability issues for CMOS devices. Many studies classify stress induced leakage current (SILC) modes in GOBD in three categories: A, B, and C-mode SILC [105],[106]. A-mode SILC is induced by trap-assisted tunneling mechanisms where electrons pass from the cathode to the anode via defect sites (neutral traps) in the SiO_2 by the electrical field [107],[108]. A-mode SILC degrades into B-mode SILC when the oxide experiences partial breakdown, also known as soft breakdown (SBD) [105],[109]. When the oxide fails to operate as an oxide, this corresponds to C-mode SILC, which is hard breakdown (HBD) [110].

Experimental observations indicate that the mean time to failure is a function of the total gate oxide surface area, temperature, and gate voltage due to the weakest-link character of oxide breakdown [111]. However, when abstracting this relationship to the system level, it is important to take into account details of circuit operation, not just the surface area. Moreover, circuits have been known to operate during breakdown. In order to model circuit performance degradation under breakdown, time dependent resistance models [23],[112] and time dependent leakage current models [113] have been proposed for SPICE simulation.

In order to represent SBD and HBD in time-dependent dielectric breakdown, we model the oxide breakdown resistance as function of time with the percolation and quantum point contact (QPC) model [34],[35]. The percolation model (PM) 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 [114], as shown in Figure 3.17. During electrical stress on the gate, the trap density in the oxide

increases [106]. Figure 3.18 shows the probability plot of the time for percolation paths to develop. SBD leakage resistance (R_{SBD}) is calculated based on the number of paths and QPC model [115].

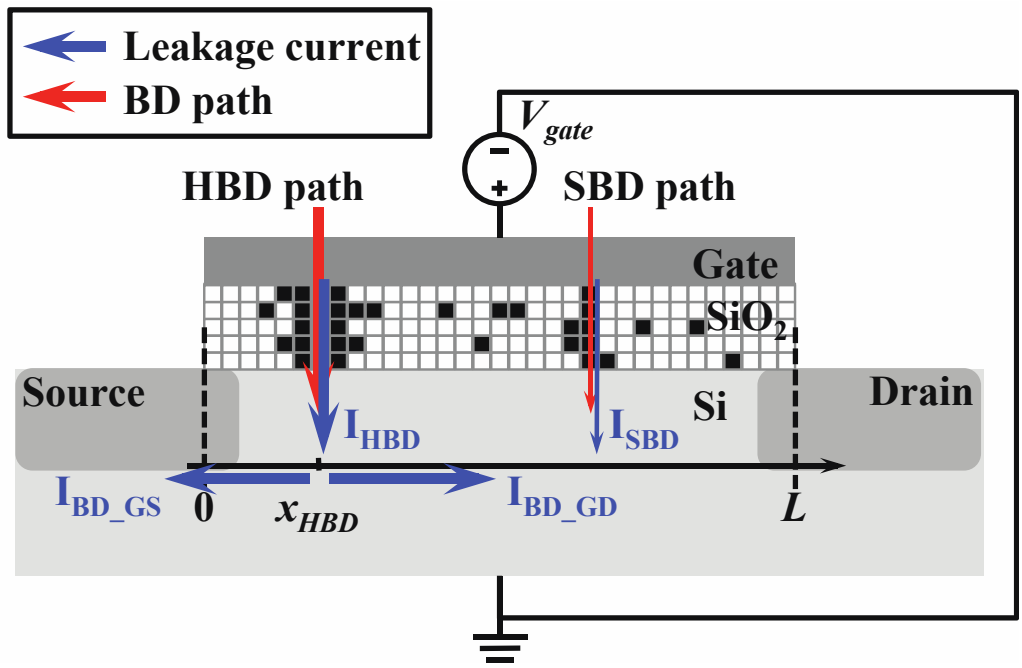
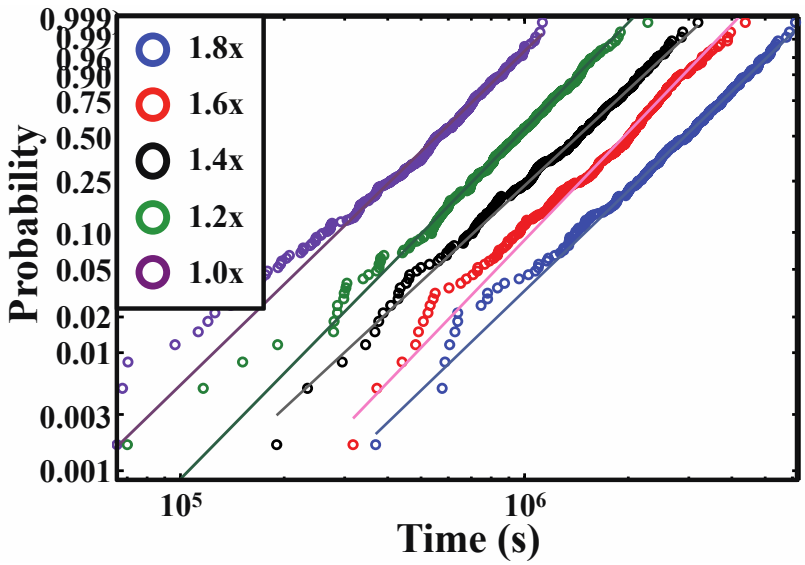
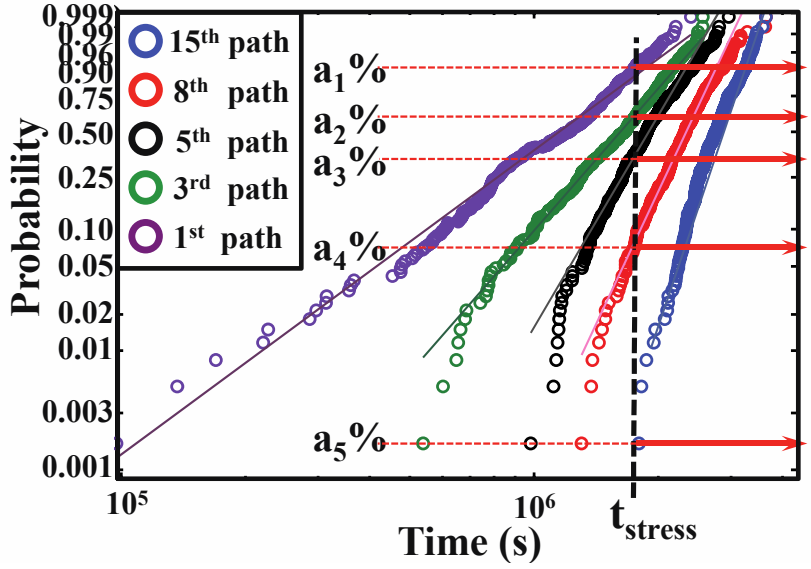


Figure 3.17: Defect generation in the SiO_2 layer based on a 2D percolation model for SBD and HBD paths.



(a)



(b)

Figure 3.18: Time distribution of defect generation in SiO_2. (a) The probability distribution of the time of occurrence of the 1st SBD path for different gate sizes. (b) The probability distribution of the number of SBD paths for a fixed gate size as a function of time.

When a critical density of traps is reached, catastrophic failure, known as HBD, occurs [106]. We use the voltage- dependent power-law gate oxide degradation model for HBD [116] to calculate the time dependent resistance values for each MOSFET under HBD.

CHAPTER 4

AGING ASSESSMENT FRAMEWORK

Because the wearout mechanisms are activity and temperature dependent, our methodology includes determining the temperature and stress for each device while running benchmarks. A framework for the acquisition of spatial and temporal thermal/electrical stress of the system was constructed.

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 months to finish simulating a single benchmark. On the other hand, simulating microprocessors with standard benchmarks on an FPGA takes only a few minutes. Our electrical aging assessment framework is schematically described in Figure 4.1, which provides an efficient way to acquire electrical and thermal profiles for any digital system for use in system-level reliability analysis.

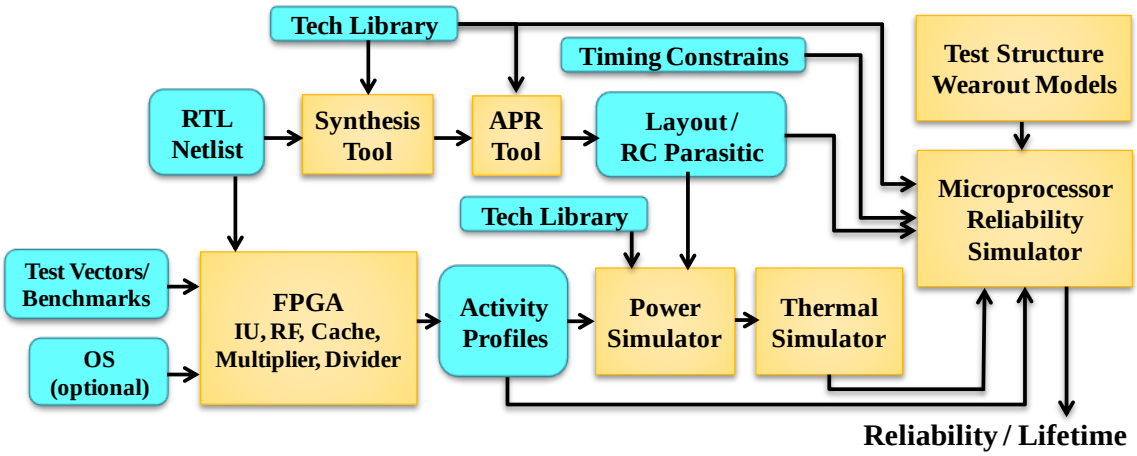


Figure 4.1: The schematic of the proposed electrical/thermal aging assessment framework is shown. Yellow blocks indicate tools, while blue blocks indicate data.

The RTL has been synthesized and loaded to the FPGA with the Xilinx ISE (Integrated Software Environment) [117]. Once the FPGA is programmed, the activity can be collected by placing counters at the I/O ports to track the state probabilities and the toggle rates of the ports during application runtime, as illustrated in Figure 4.2.

Since the I/O ports for each unit can be found on the top of each module, the counters are attached to the ports automatically with a scripting language. The activity transportation unit is inserted into the RTL automatically as well. The complexity of this RTL revision process is O(n), where n is the number of the number of I/O ports. Since the complexity is linear, the RTL revision process is scalable and can be implemented for large systems. Our current work focuses on implementing a microprocessor on a single FPGA, so the revised RTL is executable as long as the FPGA has enough resources (gates) to support large systems. A set of standard benchmarks [84] were used as the applications for analysis.

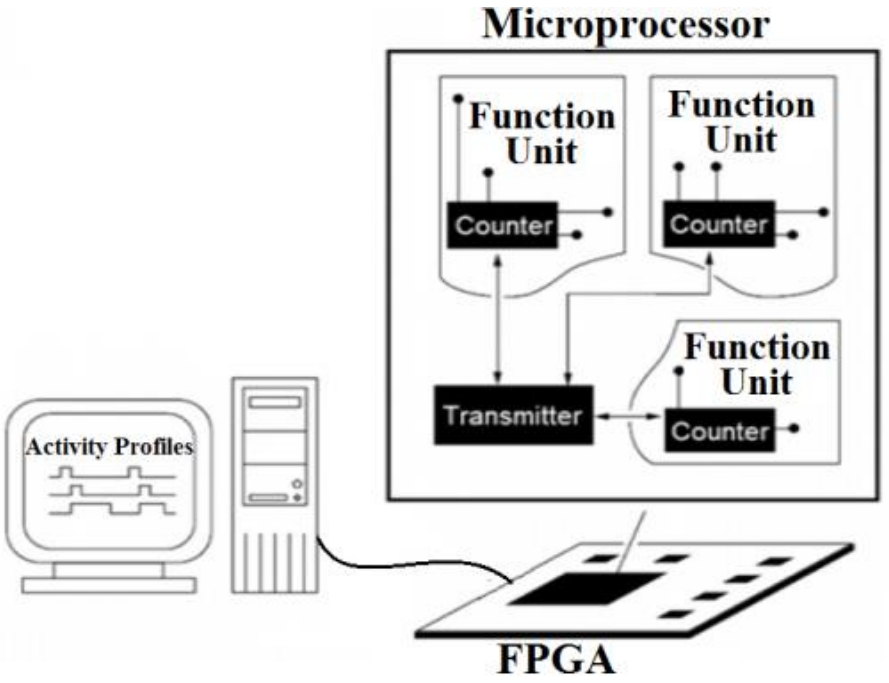


Figure 4.2: The system used to collect activity profile of microprocessor contains an FPGA board that implements the microprocessor system and exports data on the activity profile to a PC.

The I/O activities and the gate-level netlist were then used for activity propagation to each net in the design, depending on its logic behaviour, for a complete stress/transition probability profile of the internal nodes of the microprocessor under study, as illustrated in Figure 4.3. This component is done in software on a block-by-block basis. Thus, we have the probability of a transition occurring at any node and the probability at each state, i.e., the probability at logic "1". Figures 4.4 and 4.5 show the distributions of the state probability and the transition rate, respectively, when the microprocessor is running a set of standard benchmark. The distributions of the dielectric stress probability, as shown in Figure 4.6, can be further determined from equation (3.2). It can be seen that the distributions of the state probability, transition rate and dielectric stress probability are almost uniform.

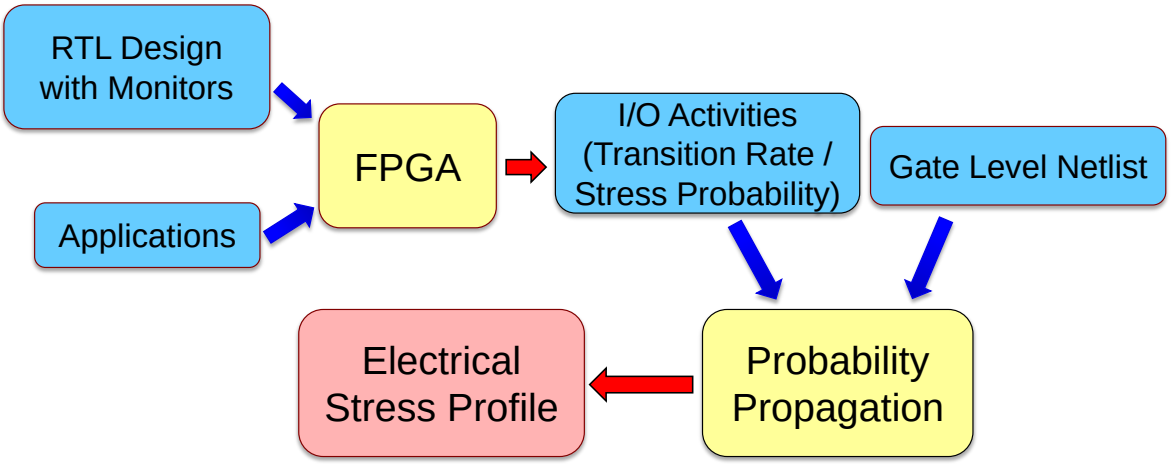


Figure 4.3: The flow of Acquisition of electrical stress profile is shown.

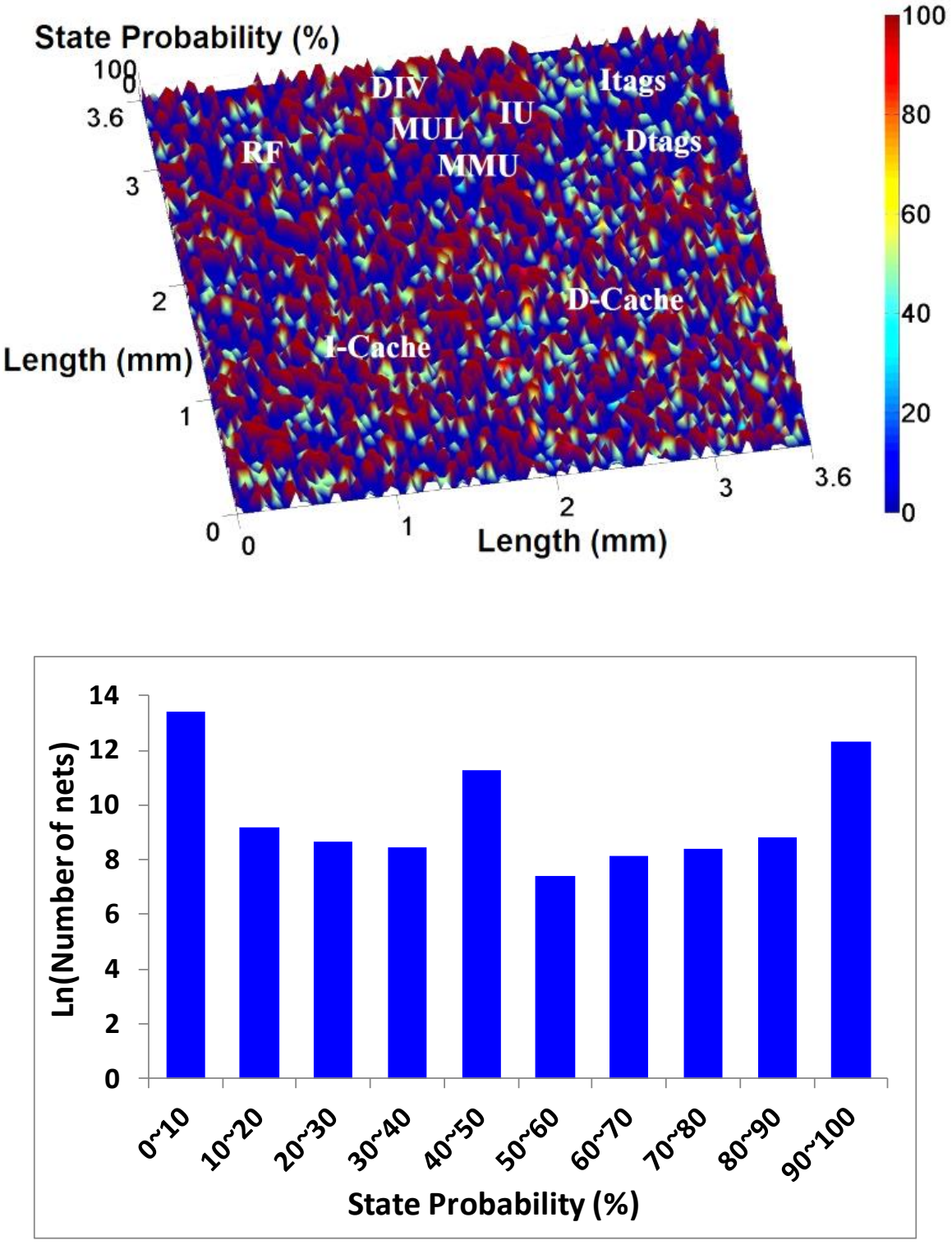


Figure 4.4: The spatial distribution of the state probability for an example microprocessor is shown while running a set of standard benchmarks.

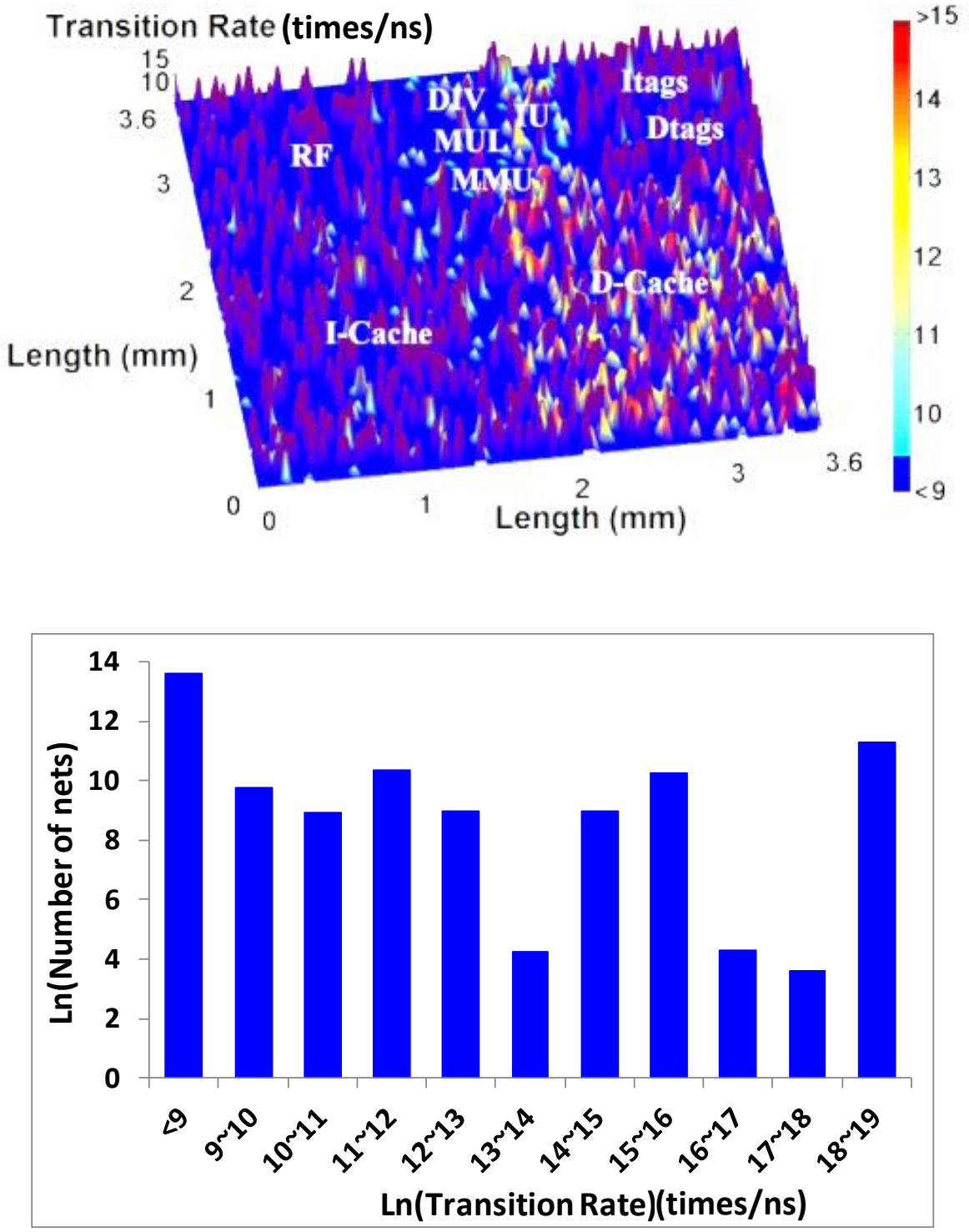


Figure 4.5: The spatial distribution of the transition rate for an example microprocessor is shown while running a set of standard benchmarks.

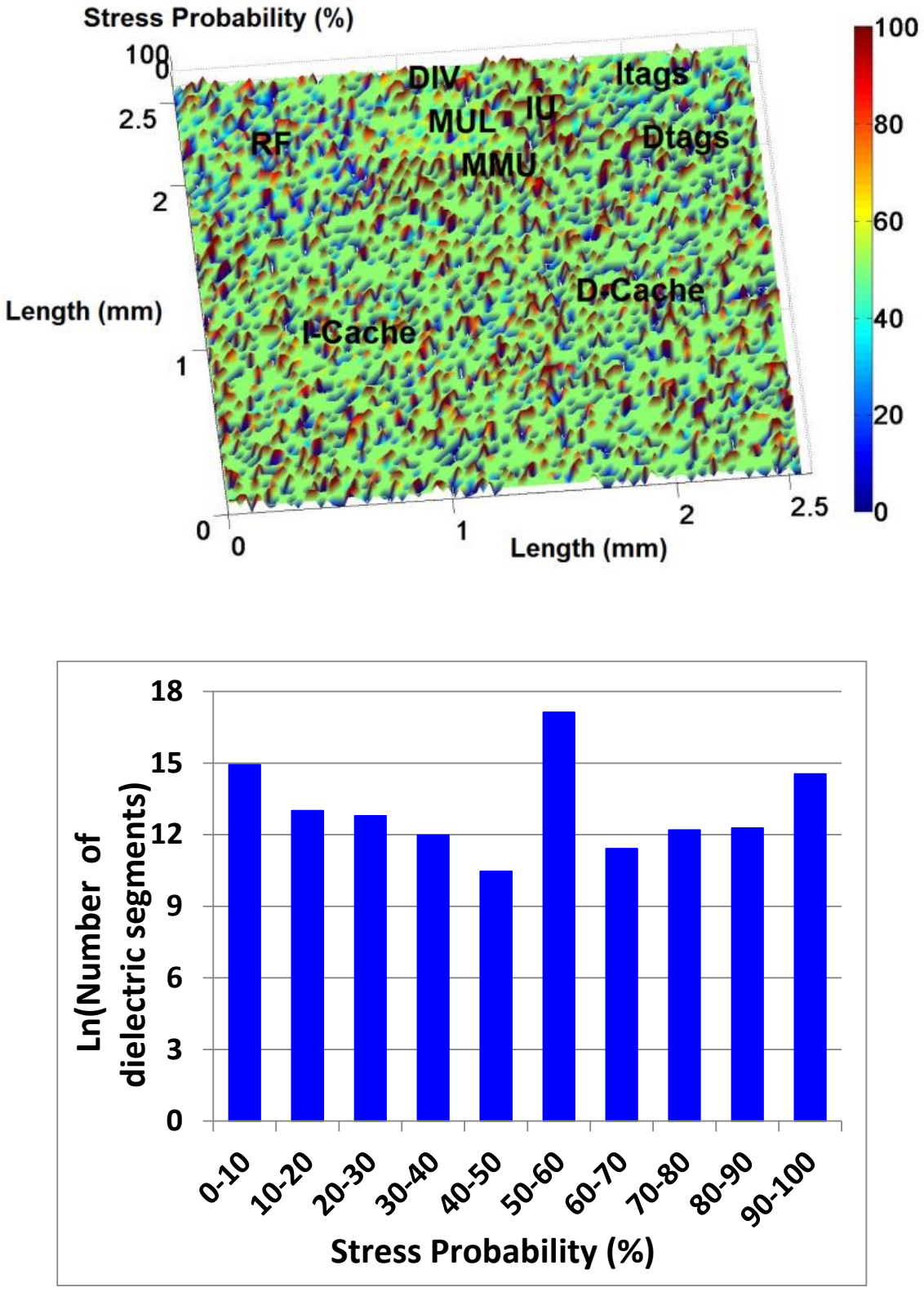


Figure 4.6: The spatial distribution of the dielectric stress probability for an example microprocessor is shown while running a set of standard benchmarks.

The propagation of transition rate and state probability was verified by comparing the exact transition numbers and state periods of randomly selected nets from the microprocessor with the ones calculated by propagations. The results, as illustrated in Figure 4.7, which shows that the percent errors for more than 90% of the selected samples are less than 10% for the transition rate and more than 80% of the selected samples have errors that are less than 15% for the state probability. The high errors are mostly from the nets in deeper locations of the circuit that are far from the I/Os. Since errors are cumulative, activity propagation to deeper stages leads to a larger difference between the real transition rate/state probability and the calculated ones.

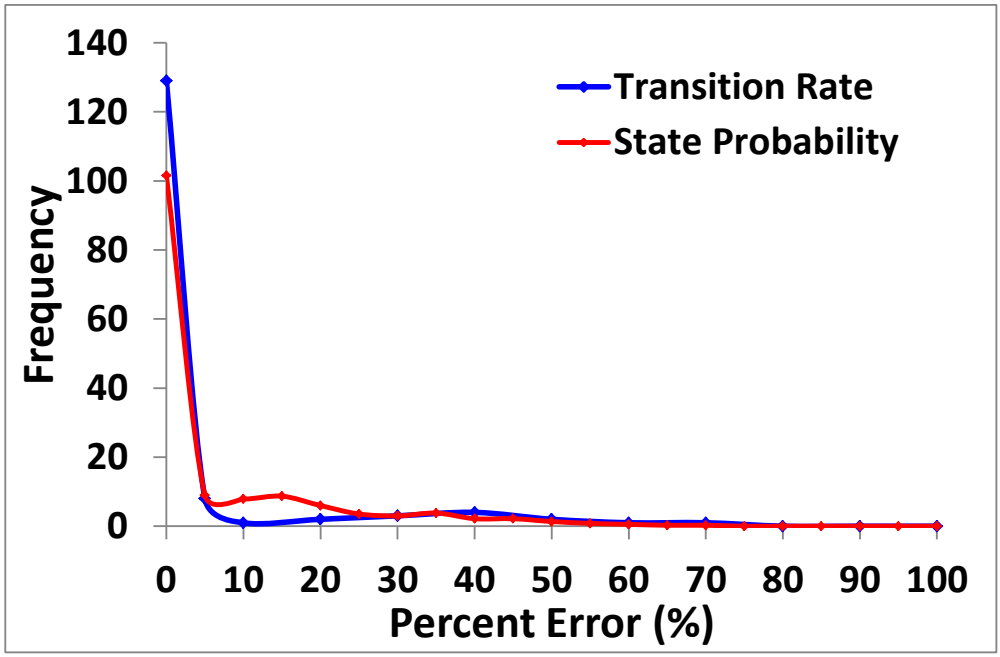


Figure 4.7: Percent error distribution of the random-selected interconnects.

The temperature variation throughout the microprocessor is also taken into account when modeling different wearout mechanisms. The netlist was used for layout

generation, as illustrated in Figure 4.8. 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 [118] and the thermal simulator [119], respectively, as illustrated in Figure 4.9.

Figure 4.10 shows the static temperature distribution when an example microprocessor system is running a set of standard benchmarks in active mode and when it is in standby mode. The static temperature is set to be the environmental temperature (30˚C) when the system is in the off mode. The static temperature is the temperature when the system reaches a stable status, when the heat generated by task execution and the heat dissipated by the cooler are balanced. None of the benchmarks that were considered in this study exhibited thermal runaway. The thermal transients associated with switching between active, standby, and off states were assumed to have a negligible impact, since the transition time was assumed to be small compared to the time in any state.

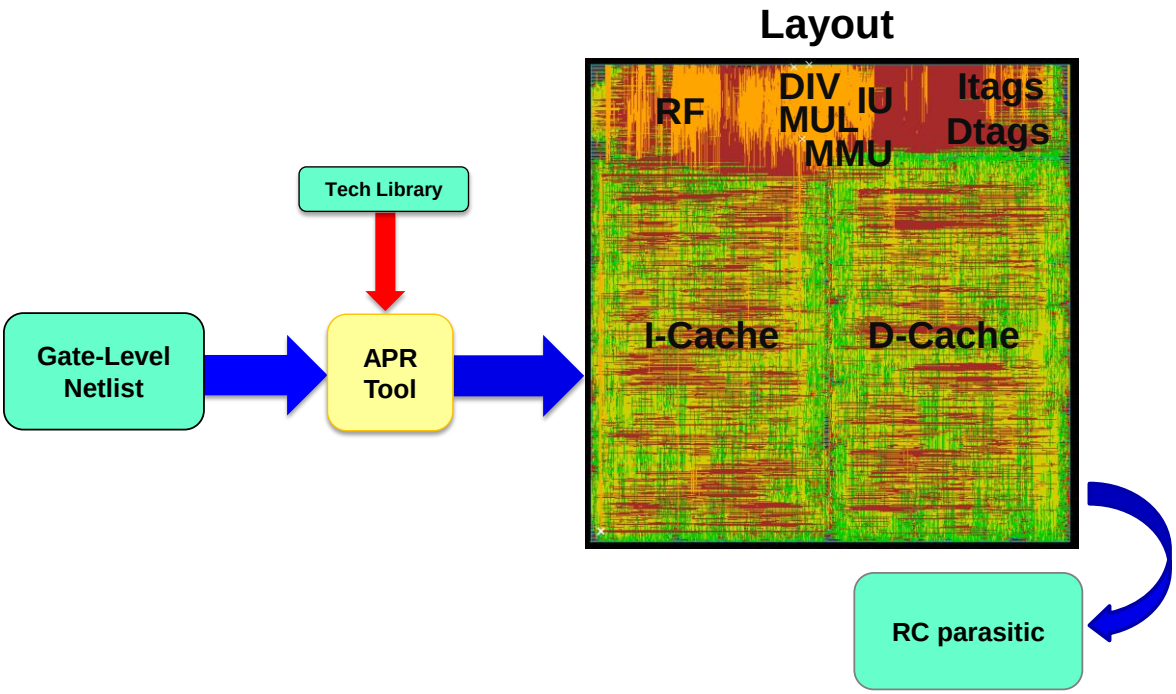


Figure 4.8: The flow of RC parasitic extraction is shown.

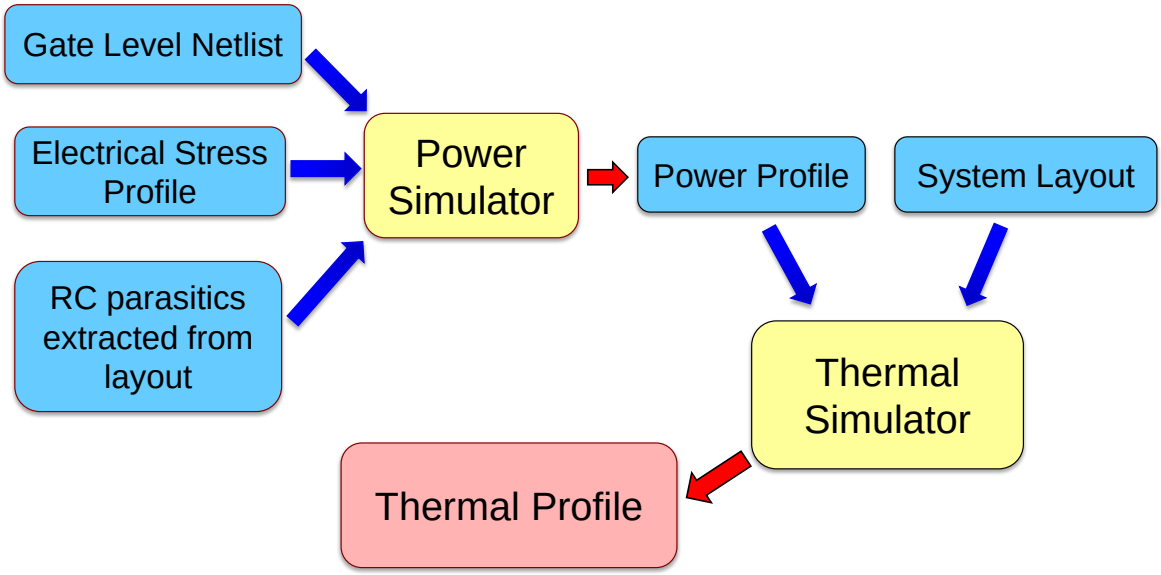
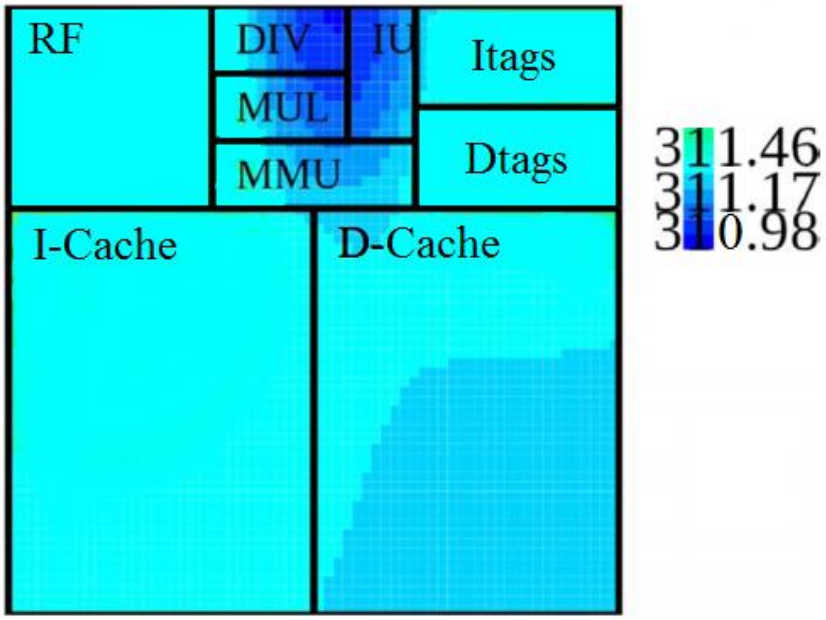
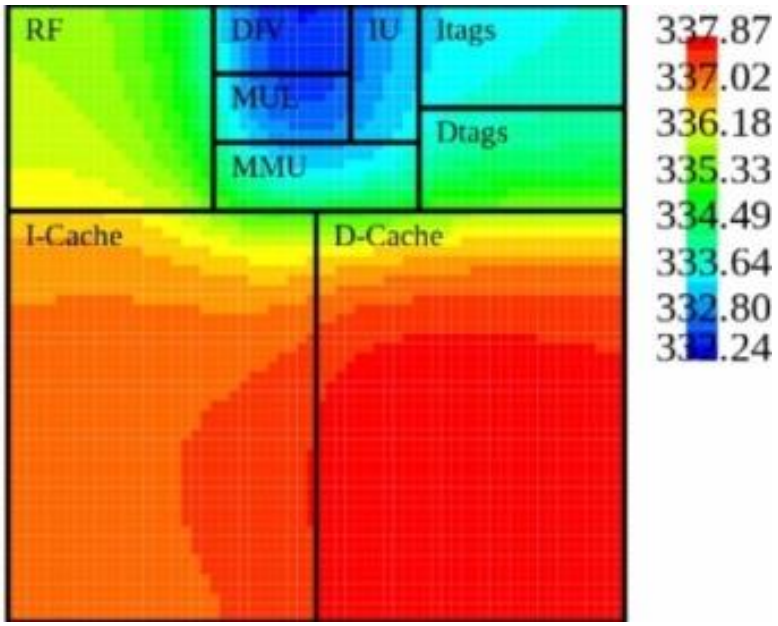


Figure 4.9: The flow of acquisition of power and thermal profiles is shown.



(a)



(b)

Figure 4.10: The static temperature distributions for an example microprocessor in (a) standby mode and (b) operation mode are shown while running a set of standard benchmarks.

Then, combining the layout, the RC parasitics, the thermal profile and the calculated probability of current flow and voltage stress, we can use device level models described in Section 3 to characterize any wearout mechanism in every feature in the layout and unit of the microprocessor under study to determine the wearout profile of the system.

CHAPTER 5

LIFETIME AND RELIABILITY ANALYSIS DUE TO BACKEND WEAROUT MECHANISMS (BTDDB, EM, SIV)

5.1 Microprocessor Lifetime Models

It should be noted that circuits wearout for a variety of reasons, both related to devices and interconnect. All of these wearout mechanisms happen simultaneously. It is common to describe backend wearout mechanisms with a Weibull distribution distribution [120],[121]

$$P(TF) = 1 - exp\left(-(TF/\eta)^{\beta}\right), \tag{5.1}$$

having two parameters: the characteristic lifetime, η, and shape parameter, β. The time-to-failure is denoted with *t*. The characteristic lifetime is the time-to-failure at the 63% probability point, when 63% of the population has failed, and the shape parameter describes the dispersion of the failure rate population. Typically, the shape parameter is close to one.

The characteristic lifetime of the microprocessor, η_{chip} , is the solution of [61]-[63]:

$$1 = \sum_{i=1}^{n}\left(\eta_{chip}/\eta_i\right)^{\beta_i}, \tag{5.2}$$

where $\eta_i, i = 1, \dots, n$ are the characteristic lifetimes of all the underlying components (interconnect segments and vias), and $\beta_i, i = 1, \dots, n$ are the corresponding shape parameters. Similarly [61],[62],

$$\beta_{chip} = \sum_{i=1}^{n}\beta_i\left(\eta_{chip}/\eta_i\right)^{\beta_i}. \tag{5.3}$$

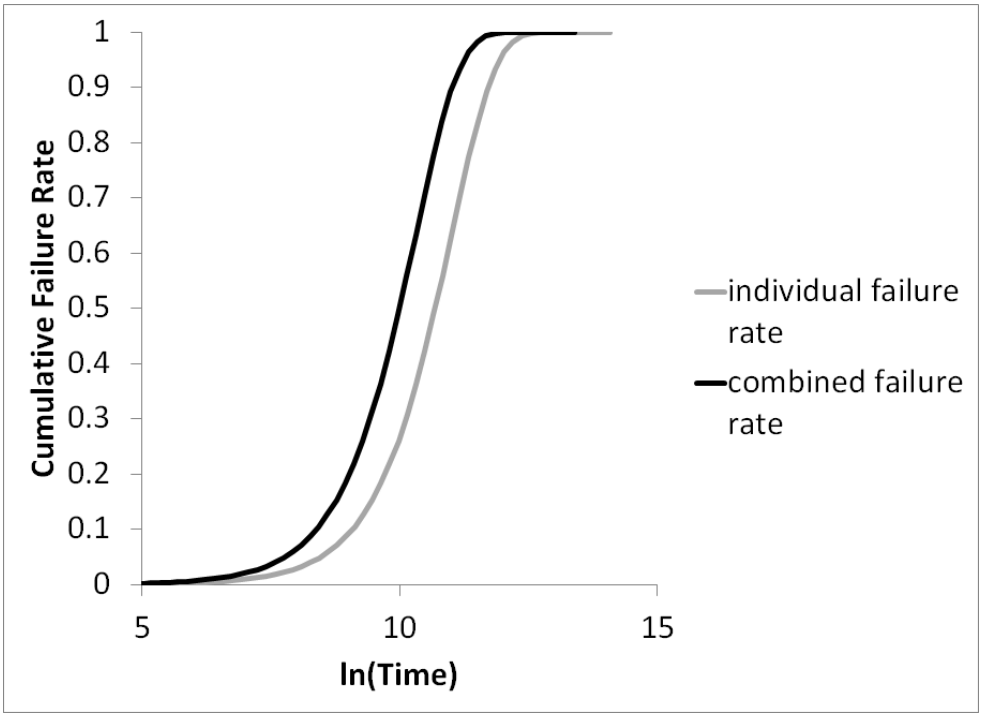
If the distribution of the lifetime of the full chip failure rate is a distribution, it is sufficient to determine the two parameters, η_{chip} and β_{chip}. These two parameters are sufficient to approximate the full chip distribution. However, the full chip failure rate distribution is not necessarily a Weibull distribution. For the exact distribution, at all probability points, *P*, one solves

$$-\ln(1-P) = \sum_{i=1}^{n}(t/\eta_i)^{\beta_i} \tag{5.4}$$

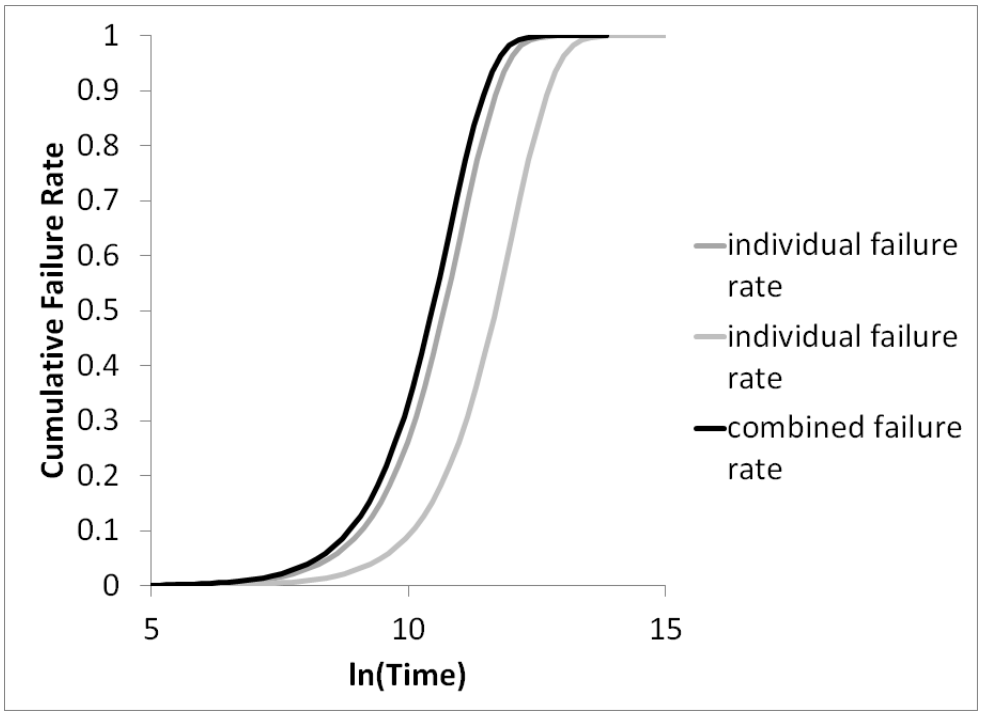
for *t*. Note that we do not need to propagate failure rate distributions through a chip. The chip is a "series" reliability system, where any failure in any component causes the full system to fail. Hence, the simulator (a) determines the characteristic lifetimes and shape parameters for all of the underlying geometries or components of each layer, accounting for temperature and use conditions with equation (3.1) for BTDDB, equation (3.13) for EM, and equation (3.14) for SIV, and (b) applies equations (5.2) and (5.3) to solve for η_{chip} and β_{chip}. Equations (5.2)-(5.4) provide a method to combine millions of component-level Weibull distributions into a single system-level full chip failure rate distribution, where equations (5.2) and (5.3) provide parameters for an approximate Weibull distribution and equation (5.4) provides the exact solution.

Consider, for example, two Weibull distributions, with the same failure rate parameters. The combined failure rate is worse than each of these failure rate distributions, as shown in Figure 5.1(a). It is clear that the failure rate for the combination of components, each described with the same Weibull parameters, is worse than each of the individual failure rate distributions. In addition, as the number of components increases, the failure rate of the system degrades. If on the other hand, the characteristic lifetime is significantly different for one distribution, then the component that fails first

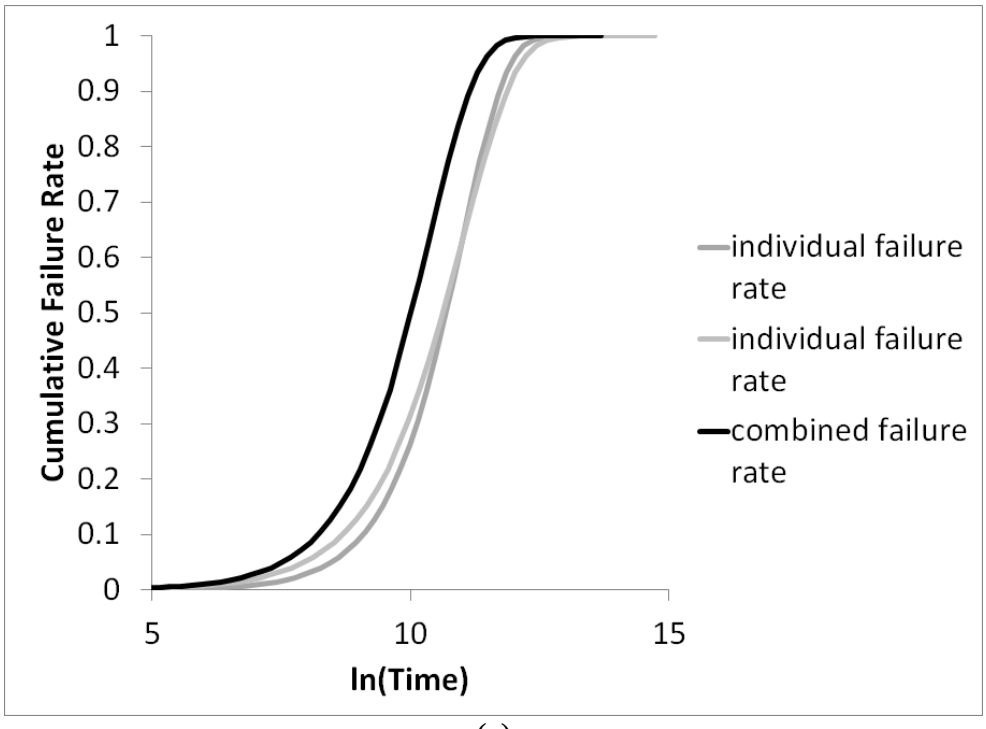
dominates (but does not completely determine) the distribution of the combined system, as shown in Figure 5.1(b). If the shape parameter is different for the two distributions, then the one with the worst shape parameter dominates the distribution of the combined system, as shown in Figure 5.1(c).



(a)



(b)



(c)

Figure 5.1: (a) Impact of combining two Weibull distributions with the same parameters. (b) Impact of combining two Weibull distributions with different characteristic lifetimes. (c) Impact of combining two Weibull distributions with different shape parameters.

Equations (5.2)-(5.4) only provide a failure rate distribution for one mode of operation. When computing the failure rate of the full system, one needs to first determine the failure rate distributions for each mode of operation by combining the component-level Weibull distributions. Then, we need to be able to combine multiple modes to provide a lifetime under use conditions. In this work, we consider the modes of operation, active, standby, and off, but the same methodology can be extended to any number of modes of operation.

Let ζ_{active} be the fraction of time in active mode. Let $\zeta_{standby}$ be the fraction of time in standby mode. And, let $\zeta_{off} = 1 - \zeta_{active} - \zeta_{standby}$ be the fraction of time in the off state. Let the active mode Weibull parameters be η_{active} and β_{active}. Similarly, the standby mode Weibull parameters are $\eta_{standby}$ and $\beta_{standby}$.

The impact of multiple operation modes is a change in the failure rate per unit time of the full system. For a Weibull distribution, $h(t) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1}$ is the number of failures per unit time, divided by the number of remaining units. For our system, involving multiple Weibull distributions,

$$h(t) = \sum_{i=1}^{n} \frac{\beta_i}{\eta_i}\left(\frac{t}{\eta_i}\right)^{\beta_i - 1}. \tag{5.5}$$

Therefore, for multiple modes of operation,

$$h(t) = \zeta_{active} \sum_{i=1}^{n} \frac{\beta_{active,i}}{\eta_{active,i}}\left(\frac{t}{\eta_{active,i}}\right)^{\beta_{active,i}-1}$$

$$+\zeta_{standby} \sum_{i=1}^{n} \frac{\beta_{standby,i}}{\eta_{standby,i}}\left(\frac{t}{\eta_{standby,i}}\right)^{\beta_{standby,i}-1}. \tag{5.6}$$

The cumulative probability of failure is $P = 1 - e^{-\int h(t)dt}$. Hence

$$P(t) = 1 - e^{\kappa}, \tag{5.7}$$

where

$$\kappa = -\zeta_{active} \sum_{i=1}^{n} \left(\frac{t}{\eta_{active,i}}\right)^{\beta_{active,i}} - \zeta_{standby} \sum_{i=1}^{n} \left(\frac{t}{\eta_{standby,i}}\right)^{\beta_{standby,i}} \quad (5.8)$$

Equations (5.7) and (5.8) provide an exact solution for the failure rate distribution of the full system. This failure rate distribution can be approximated as a Weibull distribution,k for which we must compute the characteristic lifetime and shape parameter. The characteristic lifetime corresponds to $P = 1 - e^{-1}$. Therefore, the overall characteristic lifetime, η_{use}, is the solution of

$$1 = \zeta_{active} \sum_{i=1}^{n} \left(\frac{\eta_{use}}{\eta_{active,i}}\right)^{\beta_{active,i}} + \zeta_{standby} \sum_{i=1}^{n} \left(\frac{\eta_{use}}{\eta_{standby,i}}\right)^{\beta_{standby,i}}. \quad (5.9)$$

If β is constant, then there is closed form solution:

$$\eta_{use} = \left(\zeta_{active} \sum_{i=1}^{n} \frac{1}{\eta_{acitve,i}{}^{\beta}} + \zeta_{standby} \sum_{i=1}^{n} \frac{1}{\eta_{standby,i}{}^{\beta}}\right)^{-1/\beta}. \quad (5.10)$$

Figure 5.2 shows the impact of combining two distributions with different failure rates with a varying fraction of time in active mode and the remaining time in standby mode.

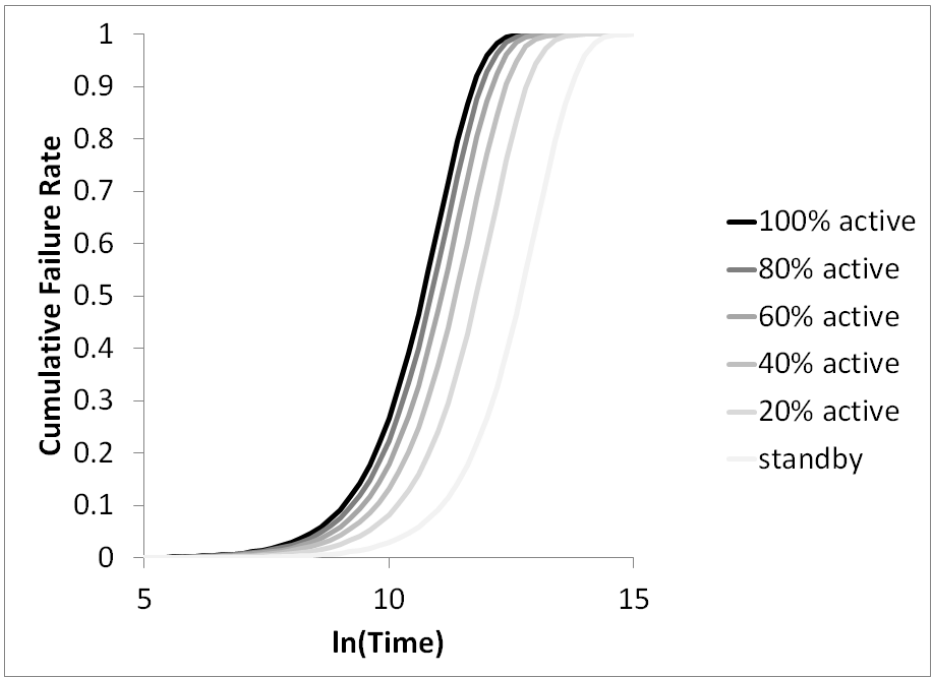


Figure 5.2: Impact of combining two Weibull distributions with different failure rates.

5.2 Lifetime Estimations for The Systems

For estimating system lifetimes, we have considered two case studies: LEON3 IP core processor [122] and the 32-bit RISC microprocessor [123]. The simulation results are presented in the following sections.

5.2.1 Case Study 1: LEON3 microprocessor

The well-known open-source LEON3 IP core processor with superscalar abilities [122] was studied. 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 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). The microprocessor includes around 240k gates.

The electrical stress and thermal profiles for the system were collected using the framework described in Section 4. The electrical and thermal profiles, together with the lifetime models from Section 3, were then used to estimate the lifetime of each functional unit in the microprocessor system. Since the microprocessor lifetimes are workload dependent, different use scenarios such as corporate, gaming, office work and general usage are also taken into account for the lifetime estimations. These realistic use conditions, as summarized in Figure 1.3.

For BTDDB, by weighting the lifetimes of operation, standby and off mode in accordance with Figure 1.3, we have estimated the lifetime of each unit within the microprocessor and analyzed the lifetime for each metal layer in the design technology used under different use scenarios, as shown in Figures 5.3 and 5.4. Figures 5.3 and 5.4 report the characteristic lifetimes, η. The characteristic lifetime is the probability point when 63% of the population has failed. When the characteristic lifetime is combined with the shape parameter, β, we have a complete probability density distribution, given by equation (5.1), provided that the resulting distribution is Weibull. Otherwise, the complete probability density distribution requires the solution of equations (5.7) and (5.8). For the sake so simplicity, in this section only the characteristic lifetime is reported since it provides an indication of the relative lifetime of different units.

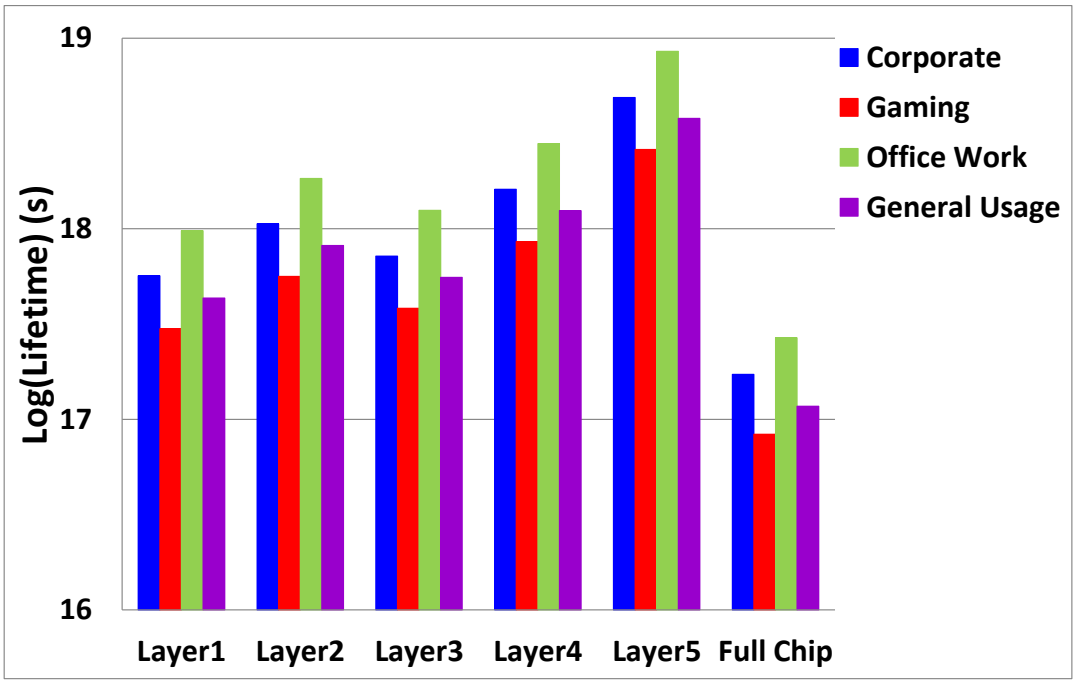


Figure 5.3: Characteristic lifetimes under different scenarios for each layer of LEON3 microprocessor due to BTDDB indicate the most vulnerable layer.

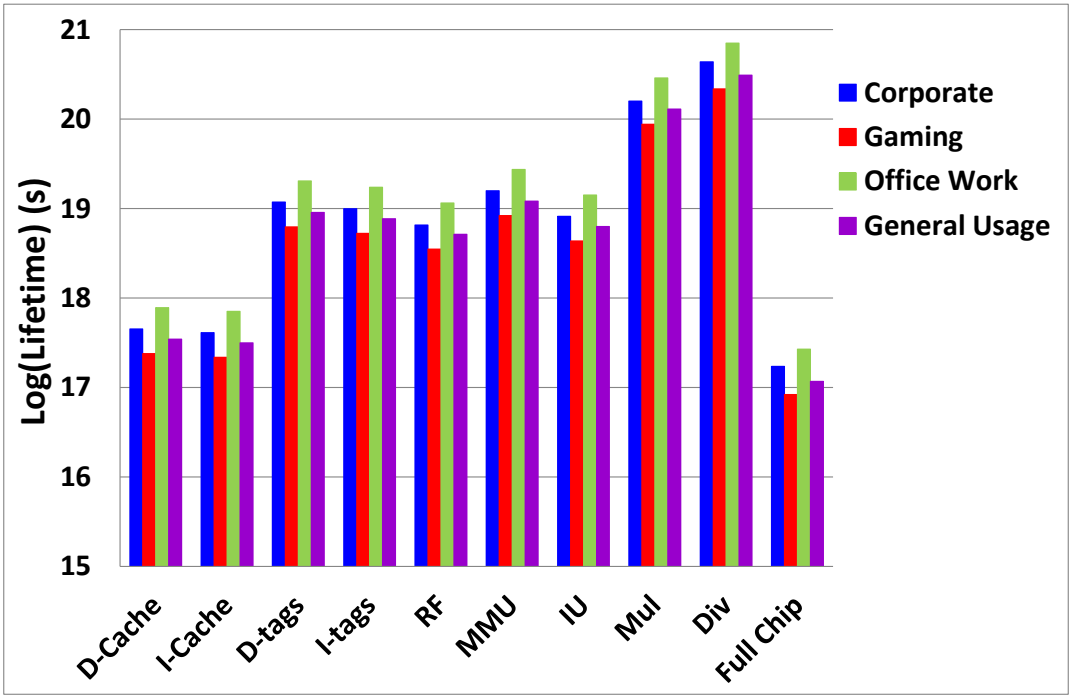


Figure 5.4: Characteristic lifetime results under different use scenarios for each unit in LEON3 microprocessor due to BTDDB indicate the most vulnerable blocks.

The lifetime of the microprocessor system under BTDDB is clearly limited by the Metal 1 layer. As we move up in the metal layer stack, the metal spacing increases, resulting in an increased time-to-failure. Our analysis shows that the data cache and the instruction cache are the lifetime-limiting units in the microprocessor. Figures 4.10, 5.3, and 5.4 also clearly suggest a strong temperature dependence of functional unit lifetimes. Among the combinational blocks, lifetime is limited by the MMU and the IU, while the MUL and the DIV blocks had relatively better lifetimes.

The microprocessor system lifetime was also investigated under EM. The results for the expected lifetimes of the microprocessor and each unit under EM are shown in Figure 5.5. The lifetime limiter is expected to be the data cache under EM. A comparison of these results with the activity and thermal profiles shown in Figures 4.5 and 4.10, respectively, indicates the strong activity and temperature dependence of functional unit lifetimes. Among the logic units, the IU is expected to have the shortest lifetime under EM.

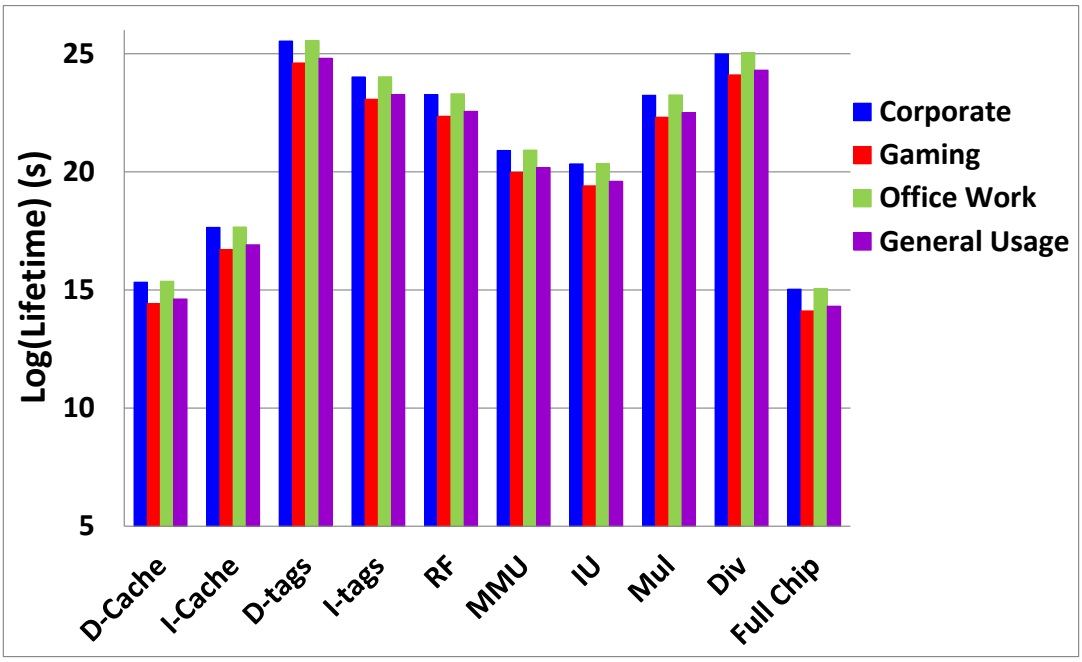


Figure 5.5: Characteristic lifetime results under different use scenarios for each unit in the microprocessor system due to EM indicate the most vulnerable blocks.

The microprocessor system lifetime under SIV was also analyzed. The results for the expected lifetimes of the microprocessor and each unit under SIV are shown in Figure 5.6. The results for SIV for the microprocessor system indicate that the system lifetime is limited by the data cache and the instruction cache. SIV is a function of temperature. A comparison of the results in Figure 5.6 with the thermal profiles shown in Figure 4.10 indicates a strong temperature dependence of the functional unit lifetimes. Among the logic units, the memory management unit is expected to have the shortest lifetime under SIV.

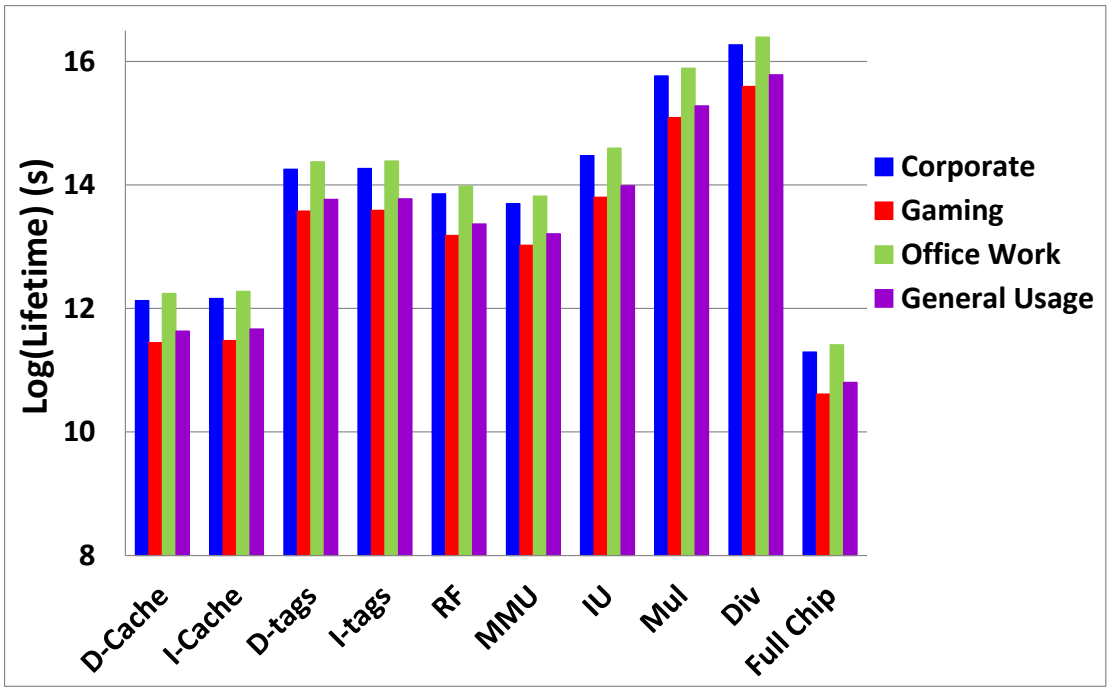


Figure 5.6: Characteristic lifetime results under different use scenarios for each unit in the microprocessor system due to SIV indicate the most vulnerable blocks.

Comparing the use scenarios, it can be seen that gaming and general usage result in the worst lifetimes, while corporate usage and office work give the best results for all backend wearout mechanisms. These use scenarios spend less time in active mode.

By comparing Figures 5.4-5.6, it can be seen that the two large blocks, the D-Cache and the I-Cache, have the most significant impact on the lifetime. To determine if this result is just due to area, we created an artificial example, where all the blocks have the same area, but vary in activity, stress, and temperature, in accordance with the use scenarios in Figures 4.5, 4.6, and 4.10. The results are shown in Figures 5.7-5.9. The results show that the caches are still limiting for lifetime, even when controlling for area. This is most likely due to their higher temperature. For BTDDB, when controlling for area, all blocks except MUL and DIV have a similar lifetime to the caches. For EM and

SIV, when controlling for area, only the IU and MMU have a similar lifetime to the caches. These two units experience a higher transition rate.

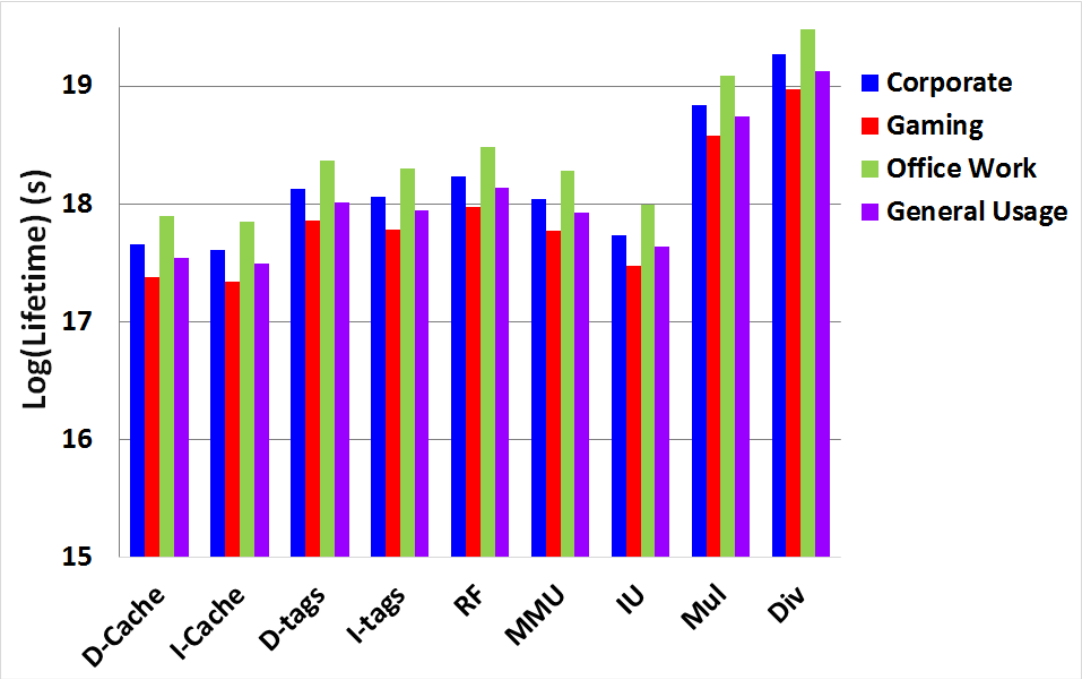


Figure 5.7: Characteristic lifetime results under different use scenarios due to BTDDB for each unit in LEON3 microprocessor where each unit is expanded so that each unit has a fixed area.

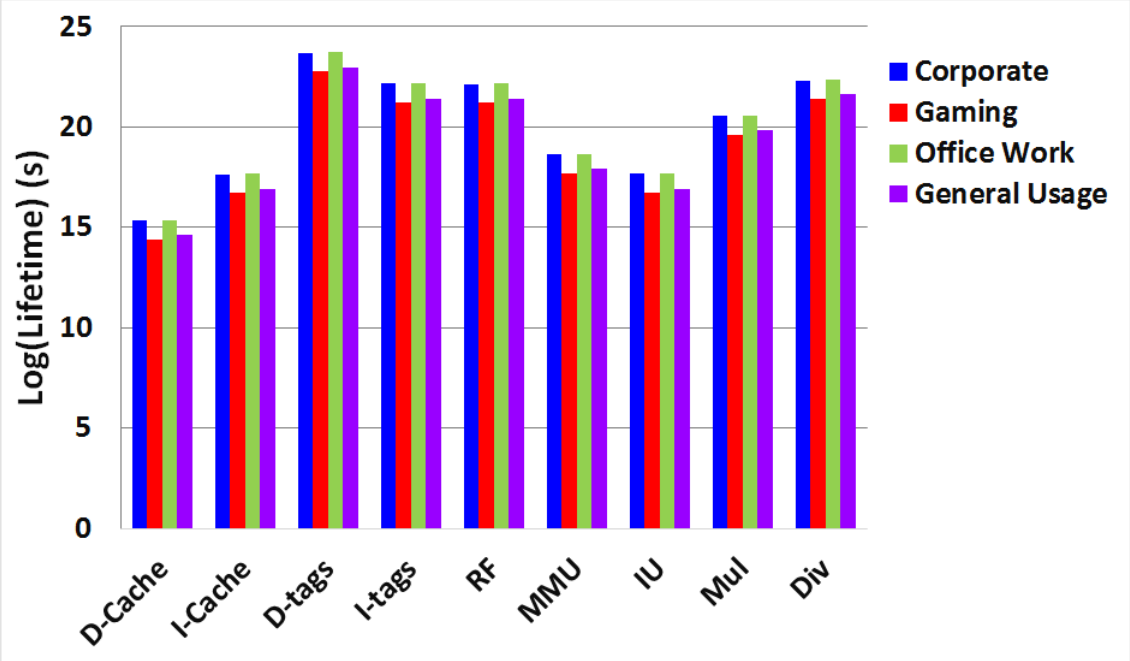


Figure 5.8: Characteristic lifetime results under different use scenarios due to EM for each unit in LEON3 microprocessor where each unit is expanded so that each unit has a fixed area.

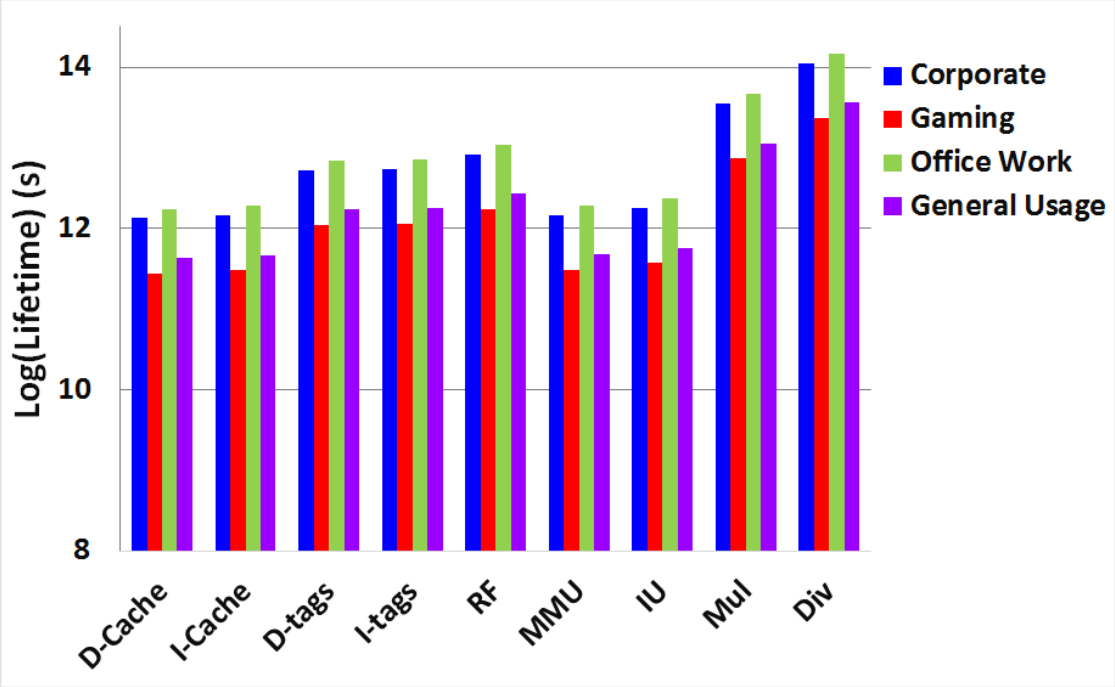


Figure 5.9: Characteristic lifetime results under different use scenarios due to SIV for each unit in LEON3 microprocessor where each unit is expanded so that each unit has a fixed area.

We also considered on-line reconfiguration through redundancy allocation. Seven additional columns were considered for each of the memory units in order to implement an error correcting code scheme. The microprocessor lifetime with and without redundancy is shown in Figures 5.10-5.12. It can be seen that error correcting codes can provide at least an order of magnitude improvement in lifetime for the microprocessor system. The formulas needed to take into account redundancy are given in the Appendix A.

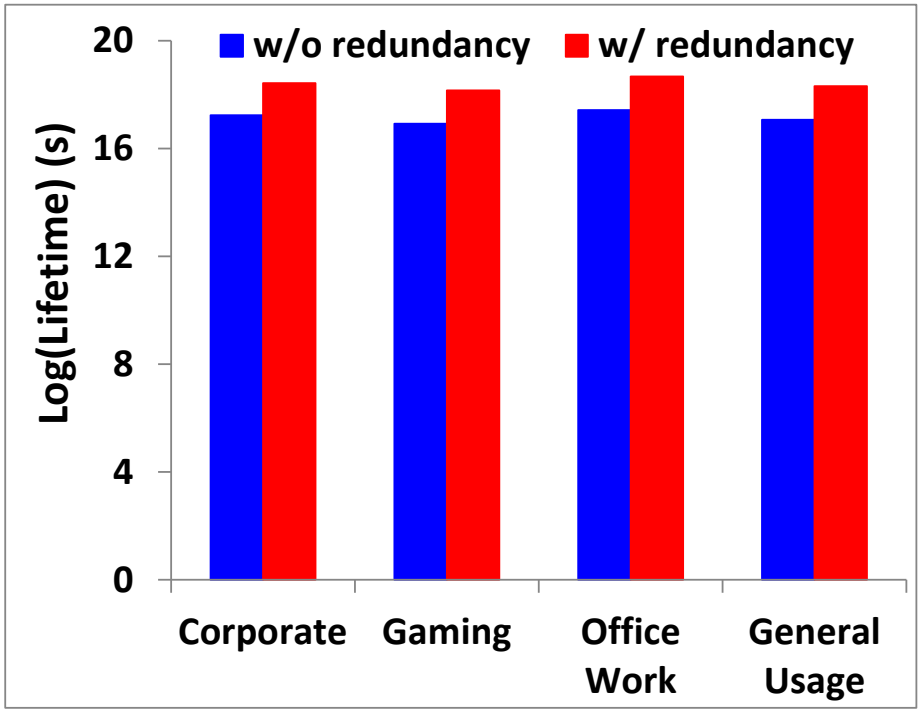


Figure 5.10: Characteristic lifetime results under different use scenarios of LEON3 microprocessor due to BTDDB with and without redundancy is shown.

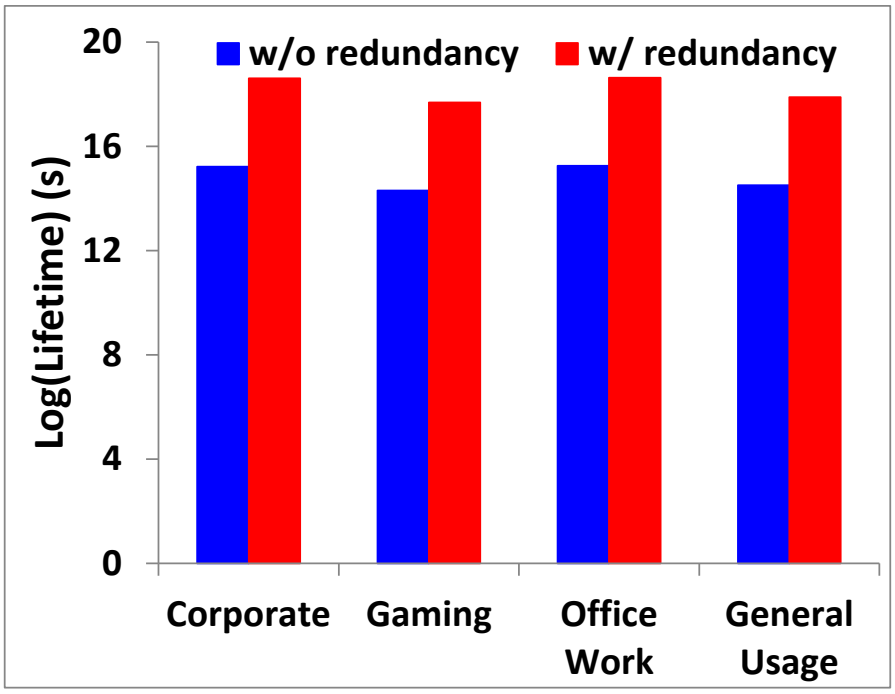


Figure 5.11: Characteristic lifetime results under different use scenarios of LEON3 microprocessor due to EM with and without redundancy is shown.

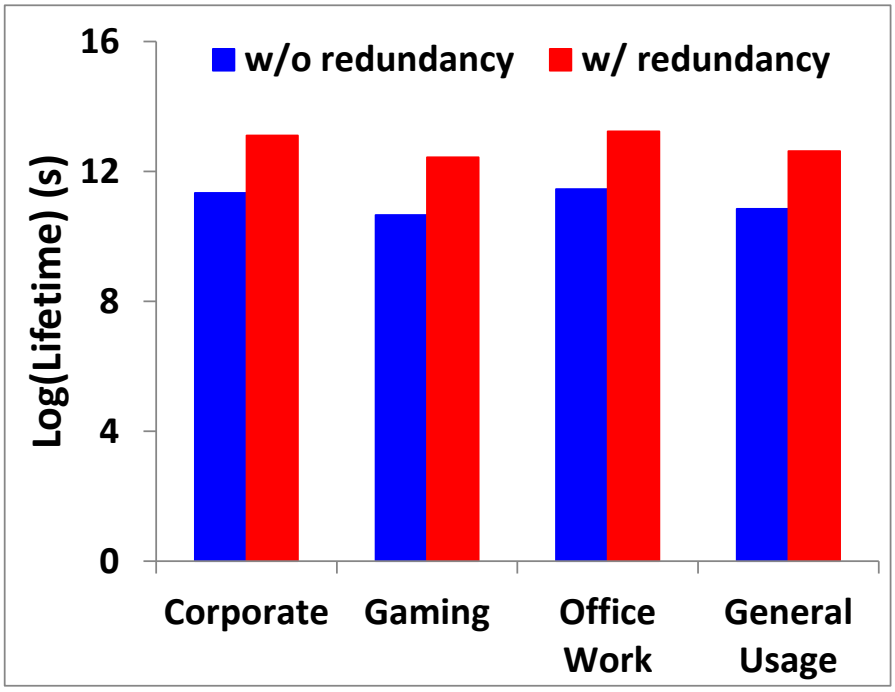


Figure 5.12: Characteristic lifetime results under different use scenarios of LEON3 microprocessor due to SIV with and without redundancy is shown.

5.2.2 Case Study 2: 32-bit RISC microprocessor

Besides LEON3, the 32-bit RISC microprocessor [123] which includes around 73k gates was also analyzed and studied.

Figure 5.13 shows the estimated lifetime due to BTDDB for each metal layer of the RISC microprocessor. Similar to the results for the LEON3, the lifetime of the microprocessor is clearly limited by the Metal 1 layer. As we move up in the metal layer stack, the metal spacing increases, resulting in an increased time-to-failure. Regarding the use scenarios, gaming has the worst lifetime, while office work has the best result.

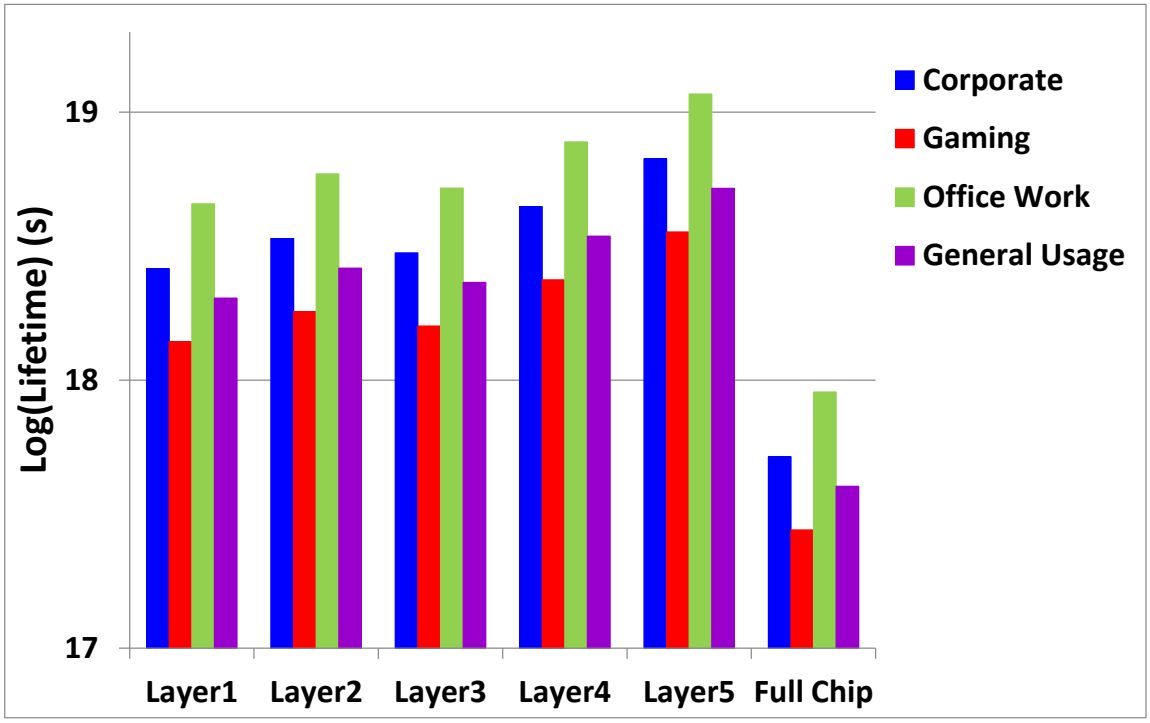


Figure 5.13: Characteristic lifetimes under different scenarios for each layer of RISC microprocessor due to BTDDB indicate the most vulnerable layer.

Figures 5.14 and 5.15 show the estimated lifetime due to EM and SIV, respectively, for the RISC microprocessor for the different use scenarios. Gaming has the worst lifetime result.

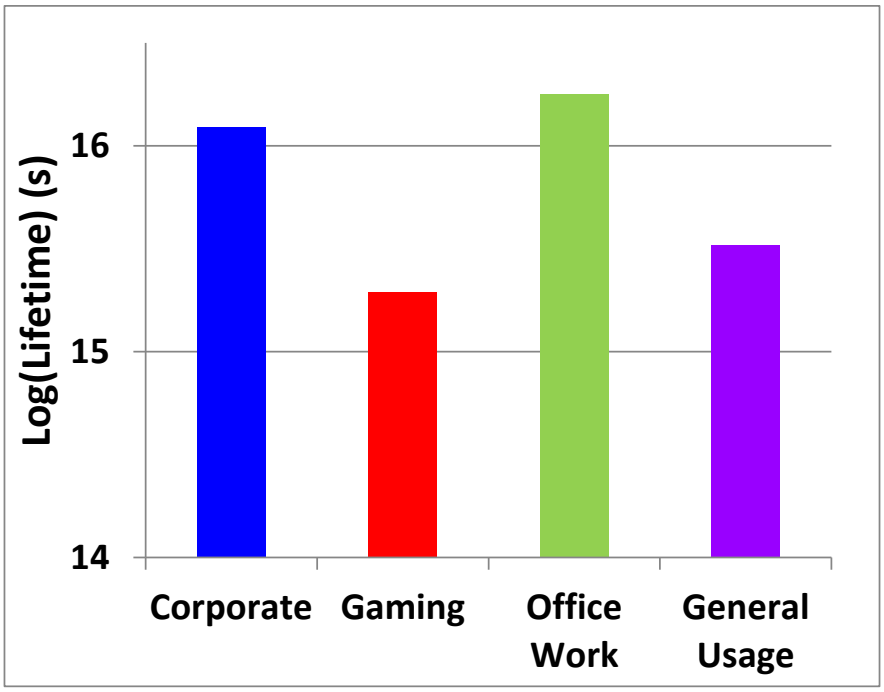


Figure 5.14: Characteristic lifetimes under different scenarios of RISC microprocessor due to EM indicate the most vulnerable layer.

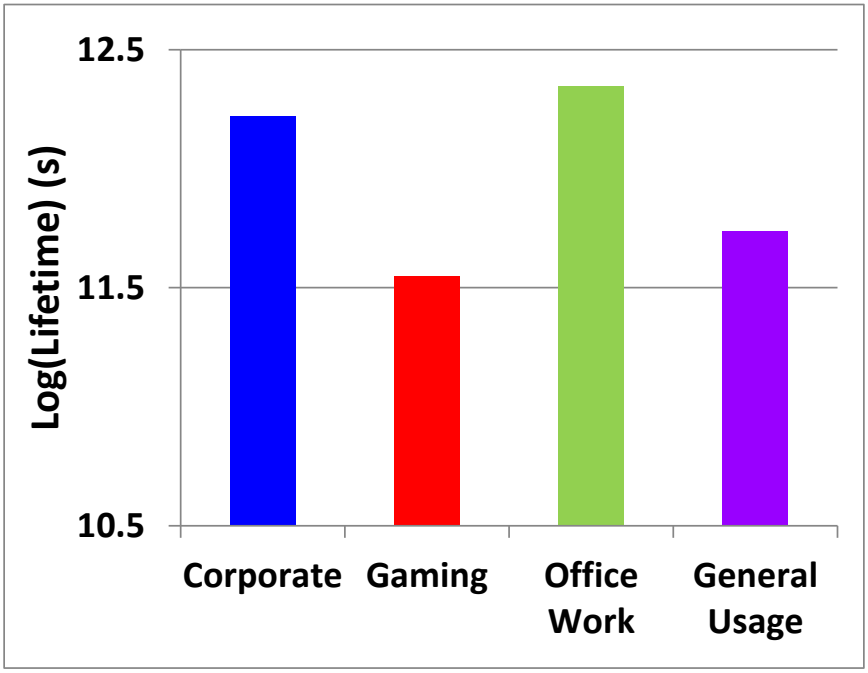


Figure 5.15: Characteristic lifetimes under different scenarios of RISC microprocessor due to SIV indicate the most vulnerable layer.

5.3 Impact of Irregular Geometries on System Lifetimes under BTDDB

To address the impact of irregular geometries on systems for BTDDB, we have studied two cases: a set of fast Fourier transform (FFT) circuits [124] with different layouts and the LEON3 IP core processor. The FFT circuits were used to study the impact of circuit geometries on dielectric lifetime, whereas the microprocessor was used to study the impact of blocks. The study of the smaller FFT circuit allows us to vary the layout and determine the impact of different geometries and design parameters during circuit synthesis, whereas the larger LEON3 circuit allows us to incorporate the impact of activity and temperature and allows us to check some of our conclusions with a larger circuit.

5.3.1 Case Study 1: FFT Circuits

Several versions of a radix-2, 256-point and 512-point FFT circuit were synthesized and implemented with the NCSU 45-nm technology library [125]. The block diagram is shown in Fig. 16. The 256-point circuit has 324-k gates and 329-k nets, and the 512-point circuit has 708-k gates and 712-k nets. The number of layers used in routing varied from five to eight. Using more routing layers results in shorter wirelength and better timing performance. Timing was optimized using buffer insertion and gate sizing.

Synopsys design compiler was used for synthesis [126]. Cadence SoC encounter was used for placement, clock-tree synthesis, routing, optimization, RCextraction [127], and static power analysis. Synopsys PrimeTime was used for timing analysis [118].

We have compared the lifetime considering only area vs. each irregular geometry in Figure 3.4 for Metal1-Metal5 for the circuit used in the study. The √E model is used to take into account the difference in design rules for each of the layers, i.e. m=1/2 in equation (3.1). The results are shown in Figure 5.16.

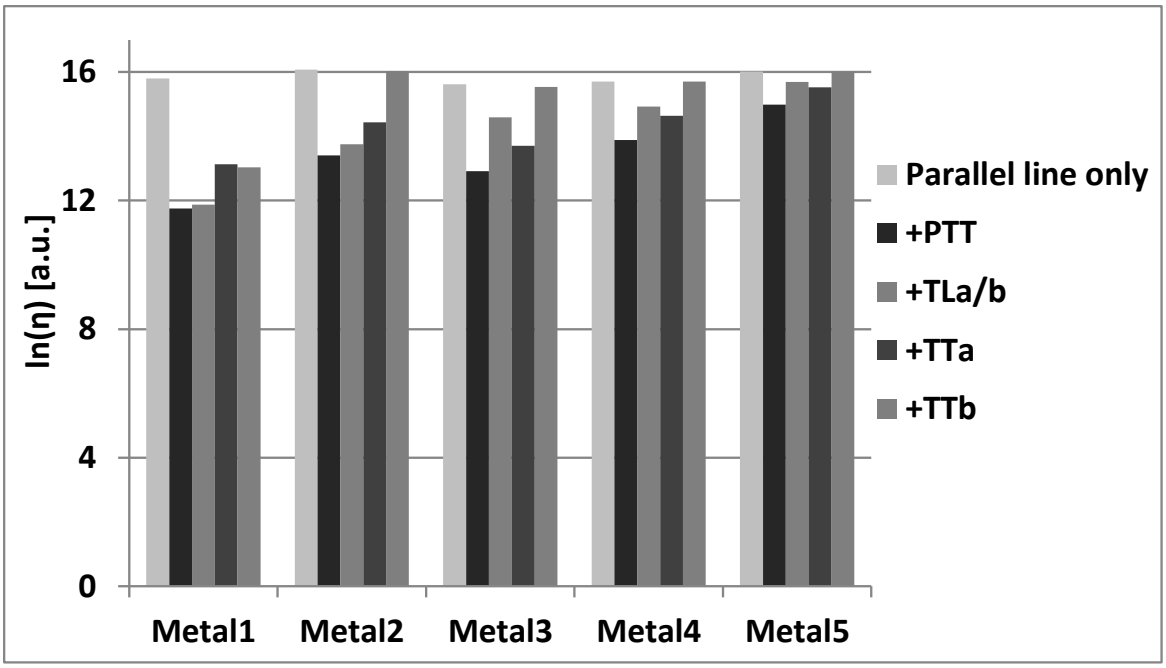


Figure 5.16: Characteristic lifetimes for individual layers of an FFT circuit considering only the dielectric between parallel lines and considering the impact of each irregular geometry separately.

We have also compared the lifetime considering only area vs. each of the irregular geometries in Figure 3.4. Figure 5.17 compares lifetimes of individual layers with and without the inclusion of degradation in lifetime due to PTT, TLa/b, TTa, and TTb for the layout of the circuit.

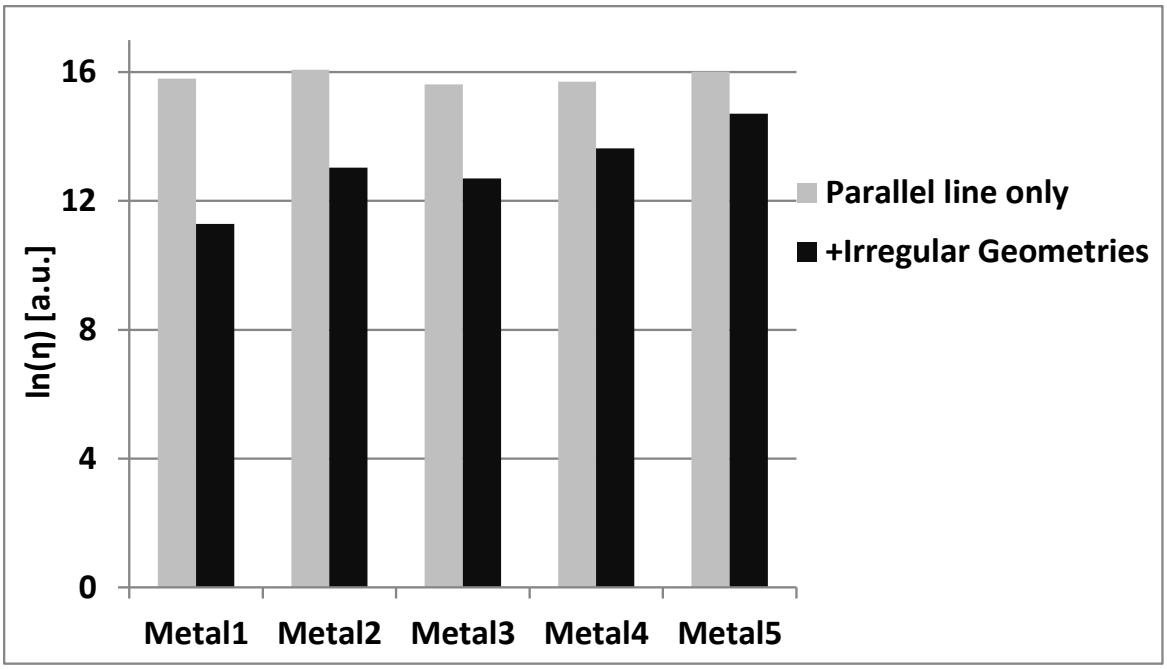


Figure 5.17: Lifetimes for individual layers of an FFT circuit considering only the dielectric between parallel lines (gray) and also considering the irregular features (black).

Figures 5.16 and 5.17 show that irregular features cause a relatively smaller difference for Metal 5 and the biggest difference for Metal 1. Also, from Figure 5.16, it can be seen that taking into account the PTT geometry impacts the lifetime of Metal 1 significantly, in comparison with considering only the area between parallel metal lines.

Figure 5.16 shows the impact of each of the irregular features. The irregular geometry that most strongly impacts lifetime is PTT. This is because there are numerous PTT geometries on Metal 1 and above. On the other hand, TTb geometries rarely (or never) occur above Metal 1, and there is a negligible impact of these geometries. TLa/b geometries essentially consist of two perpendicular wires. In general, each metal layer has a preferred routing direction, either horizontal or vertical. Perpendicular wires are usually not allowed for global routing. Therefore, TLa/b geometries above Metal 1 are

rare. TLa/b geometries were, however, frequently found on Metal 1, because Metal 1 is used in cell libraries for internal wiring.

Figure 5.17 shows the characteristic lifetime results with and without the inclusion of degradation in lifetime because of irregular features, i.e., PTT, TLa/b, TTa, and TTb. The impact of irregular features is strongest for Metal 1. This is because the number of irregular geometries decreases from Metal 1 to 5, because of routing restrictions associated with higher layers of metal.

5.3.2 Case Study 2: LEON3 microprocessor

The well-known open-source LEON3 IP core processor with superscalar abilities was studied. The electrical and thermal profiles from Section 4, together with the lifetime models from Section 3, were used to estimate the lifetime of each functional unit in the system.

Figure 5.18 shows the impact of each of the irregular geometries. The results are consistent with the FFT circuit, with PTT having the strongest impact and TTb having the least impact. Figure 5.19 shows the impact of irregular features. It can be observed that, as with the FFT circuit, the number of irregular geometries decreases for higher layers of metal.

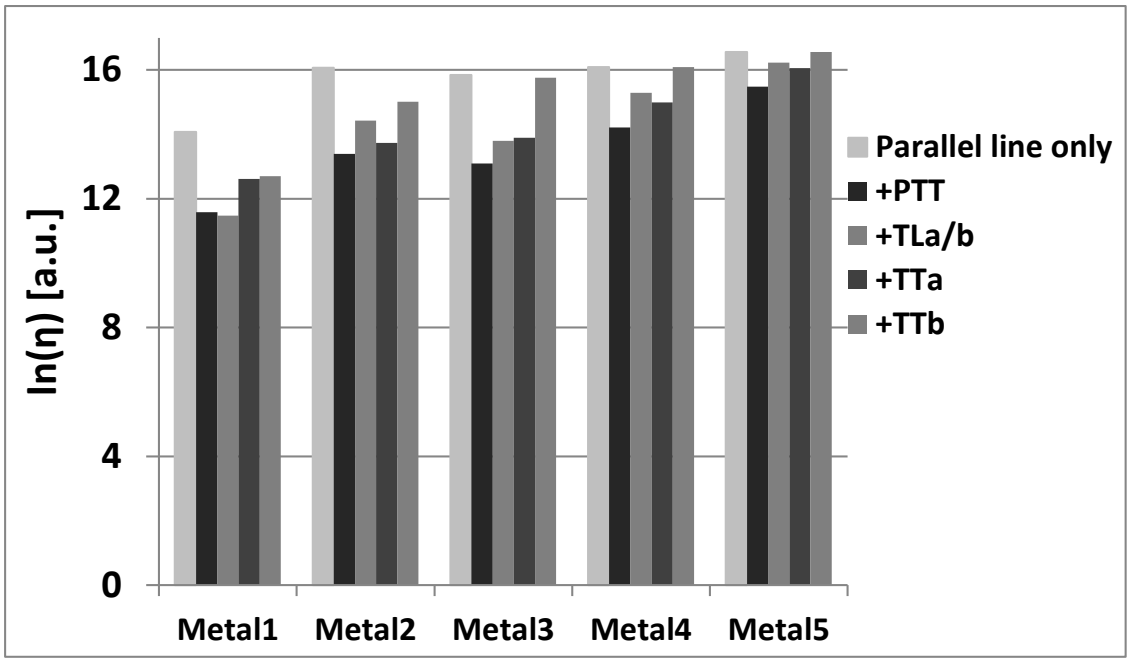


Figure 5.18: Microprocessor characteristic lifetimes for each layer considering only the dielectric between parallel lines and considering the impact of each irregular feature geometry.

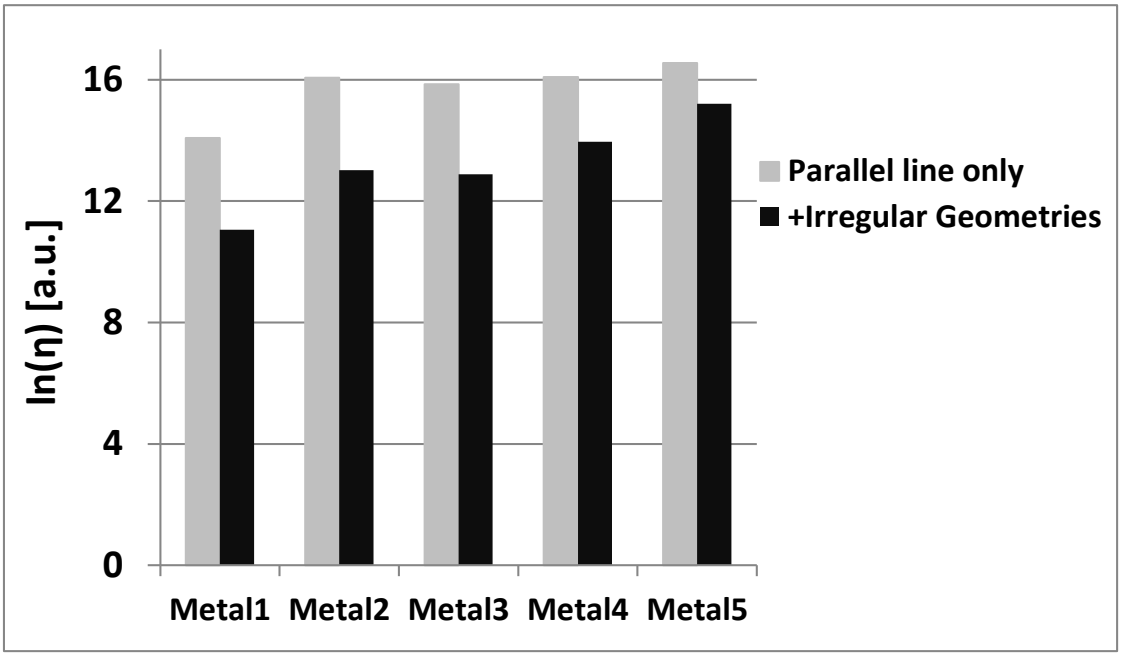


Figure 5.19: Microprocessor characteristic lifetimes for each layer considering only the dielectric between parallel lines (gray) and considering also the dielectric involved in irregular geometries.

The impact of line ends in backend dielectric TDDB was studied and found to be clearly significant. These irregular geometries can potentially impact chip lifetime and need to be separately extracted and included in a backend dielectric chip reliability simulator.

CHAPTER 6

LIFETIME AND RELIABILITY ANALYSIS DUE TO FRONTEND WEAROUT MECHANISMS (NBTI, PBTI, HCI, GOBD)

6.1 Impact of Frontend Wearout Mechanisms on Microprocessor Logic Block Reliability

6.1.1 Performance Degradation Analysis Flow

To characterize the impact of frontend wearout mechanisms on logic circuits, the signal edge degradation caused by frontend wearout mechanisms needs to be studied. Figure 6.1 illustrates the data flow and structure for modeling logic circuit performance degradation due to frontend wearout mechanisms.

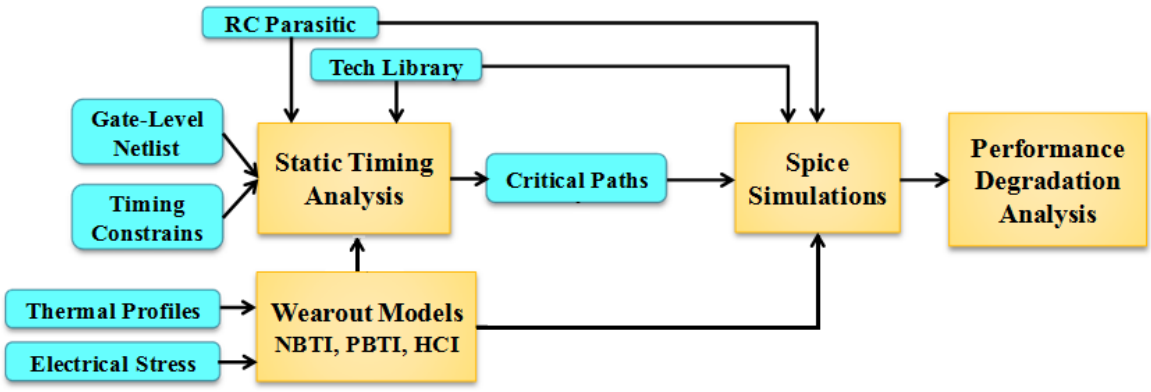


Figure 6.1: The schematic of the proposed flow for performance degradation analysis is shown. Yellow blocks indicate tools, while blue blocks indicate data.

We begin with extracting all the paths through the microprocessor system under study via static timing analysis (STA) [118].

To study the impact of frontend wearout mechanisms on the microprocessor, the BTI and HCI models described in Section 3 determine the threshold voltage drift of each device within the extracted critical paths of the microprocessor system and the TDDB models described in Section 3 determine the gate-to-source resistance (R_{G2S}) and gate-to-drain resistance (R_{G2D}) of each device. The threshold voltage drift and R_{G2S} and R_{G2D} variations are functions of the electrical stress and thermal profiles acquired from the framework illustrated in Figure 4.1. Note that the electrical stress profiles acquired from Figure 4.1 include only the activity for each net in the microprocessor. Activity propagation was implemented to obtain the electrical stress profiles for each transistor in the standard cells within the paths.

To include the additional delay caused by frontend wearout mechanisms for each gate within each path for further STA analysis, the new standard cell library has been built to model the delay drift according to the threshold voltage drifts and the R_{G2S}/R_{G2D} variations of each cell. The gate delays of the standard cells are modeled via first-order linear regression as

$$D = d_0 + \sum_{i=1}^{n} d_i \Delta X_i + d_{n+1} Slope + d_{n+2} C_{load} , \quad (6.1)$$

where D is the gate delay of a cell, n is the number of transistors in the cell, ΔX_i *is* the threshold voltage drift in transistor i (ΔV_{thi}) for BTI and HCI and the R_{G2S} and R_{G2D} variations in transistor i ($\Delta R_{G2S/G2Di}$) for GOBD, *Slope* is the input slope of the input waveform to the cell, C_{load} is the loading capacitance of the cell, d_0 is a constant term, and d_i, $i=1,2,\ldots, n+2$, are sensitivity coefficients.

Taking into account the additional delays caused by frontend wearout mechanisms, the gate-level netlist, the timing constraint file which defines system timing

speculations, and the RC parasitics, we sort all paths to find the critical ones of the microprocessor system for further analysis. As time increases and frontend wearout mechanisms degrade device characteristics, the paths are resorted to determine a new set of critical paths.

The extracted critical paths, together with all acquired RC parasitic elements in the cells and in the interconnects in the paths, are then simulated with SPICE. Process parameter variations for the devices within the extracted critical paths are taken into account in the SPICE simulations by running Monte Carlo simulations with random values for process parameters. The goal of implementing such SPICE simulations after STA is to analyze the delay of each critical path more accurately.

The threshold voltage drifts determined by the BTI and HCI models and the R_{G2S}/R_{G2D} variations determined by the TDDB models are also annotated back to the extracted critical paths to characterize the microprocessor performance degradation caused by frontend wearout mechanisms via SPICE simulations of the BTI/HCI/GOBD-induced critical paths to determine the delay of each path.

6.1.2 Logic Wearout Simulation Results

To address the impact of frontend wearout mechanisms on systems, we have studied two cases: the LEON3 IP core processor [122] and the 32-bit RISC microprocessor [123] as described in Sections 5.2.1 and 5.2.2, respectively. The simulation results are presented in the following sections.

6.1.2.1 Case Study 1: LEON3 microprocessor

A set of standard benchmarks [84] were run on the microprocessor. The microprocessor includes around 240k gates, and the runtime for executing each benchmark on the system is around one to three minutes. The electrical stress and thermal profiles for the system were collected using the framework described in Section 4. The electrical and thermal profiles, together with the lifetime models from Section 3, were then used for performance degradation analysis to analyze the impact of frontend wearout mechanisms on the microprocessor reliability.

Figures 6.2, 6.3 and 6.4 show the latency distributions of the critical paths of the microprocessor due to BTI, HCI and GOBD, respectively, for different use scenarios. The results show the critical paths of the microprocessor degrade differently while undergoing longer stress due to the frontend wearout mechanisms. The latency distributions of the critical paths not only provide us with the degradation rates of the microprocessor system due to the frontend wearout mechanisms, but also clarify the degradation variation.

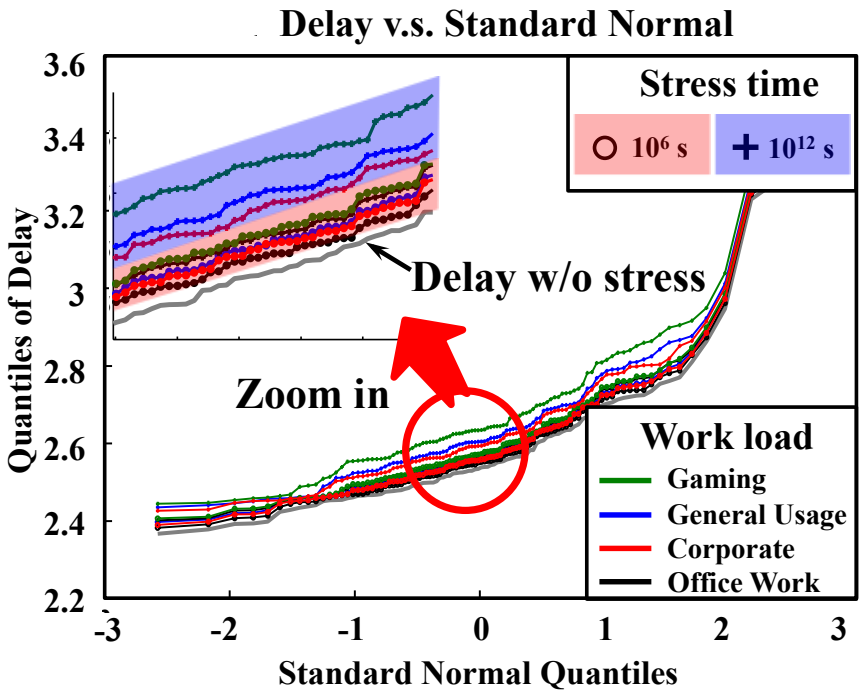


Figure 6.2: The latency distributions of the critical paths of the microprocessor due to BTI for different use scenarios and for different stress time.

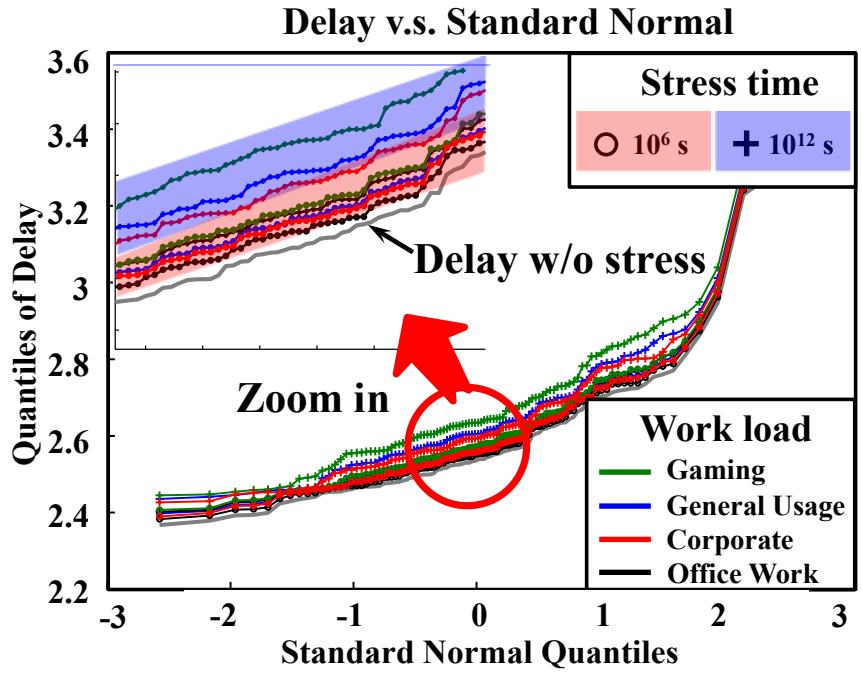


Figure 6.3: The latency distributions of the critical paths of the microprocessor due to HCI for different use scenarios and for different stress time.

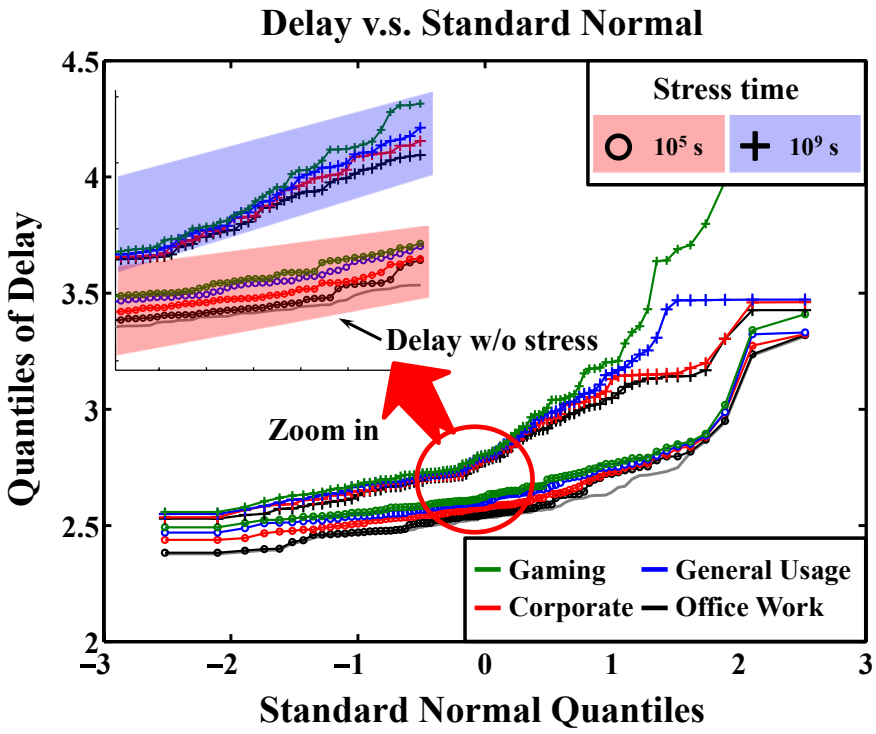


Figure 6.4: The latency distributions of the critical paths of the microprocessor due to GOBD for different use scenarios and for different stress time.

Our methodology estimates system lifetimes by analyzing degradation of critical paths of systems due to frontend wearout mechanisms dynamically based on system performance requirements. Since all the frontend wearout mechanisms result in an increase in data path delays, a system may fail when there are timing violations, such as setup time and hold time violations occurring in the critical paths. Timing violation analysis of each critical path within a system is system frequency dependent because there are bigger delay margins when a system has a lower operating frequency, and there are smaller delay margins when a system has a higher operating frequency.

For BTI, by weighting the lifetimes of operation, standby and off mode in accordance with Figure 1.3, we have estimated the lifetimes of the microprocessor under

study based on different operating frequencies for different use scenarios, as shown in Figure 6.5. The different operating modes impact the values of t_{stress} and t_{rec} in equations (3.17) and (3.20). For example, during the "off" state, t_{rec} is increased. The results clearly indicate that the estimated system lifetimes decrease as the system frequency increases, and gaming has the shortest lifetime.

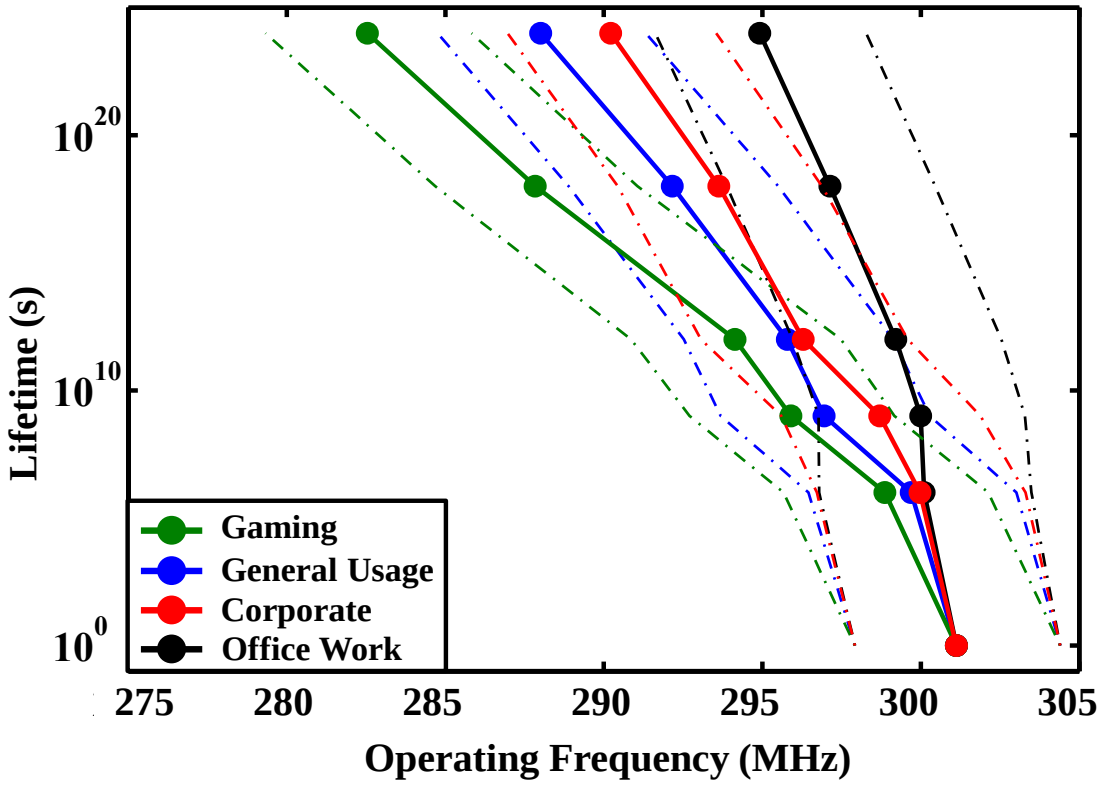


Figure 6.5: The estimated lifetimes of LEON3 microprocessor due to BTI for different use scenarios and different system frequencies. Dotted lines show the boundaries when considering process variation.

The microprocessor system lifetimes for different operating frequencies and different use scenarios were also investigated under HCI. The system lifetimes estimated by the proposed methodology are shown in Figure 6.6. Similar to BTI, the

microprocessor lifetimes estimated by our methodology decrease as the system frequency increases, and gaming has the shortest lifetime.

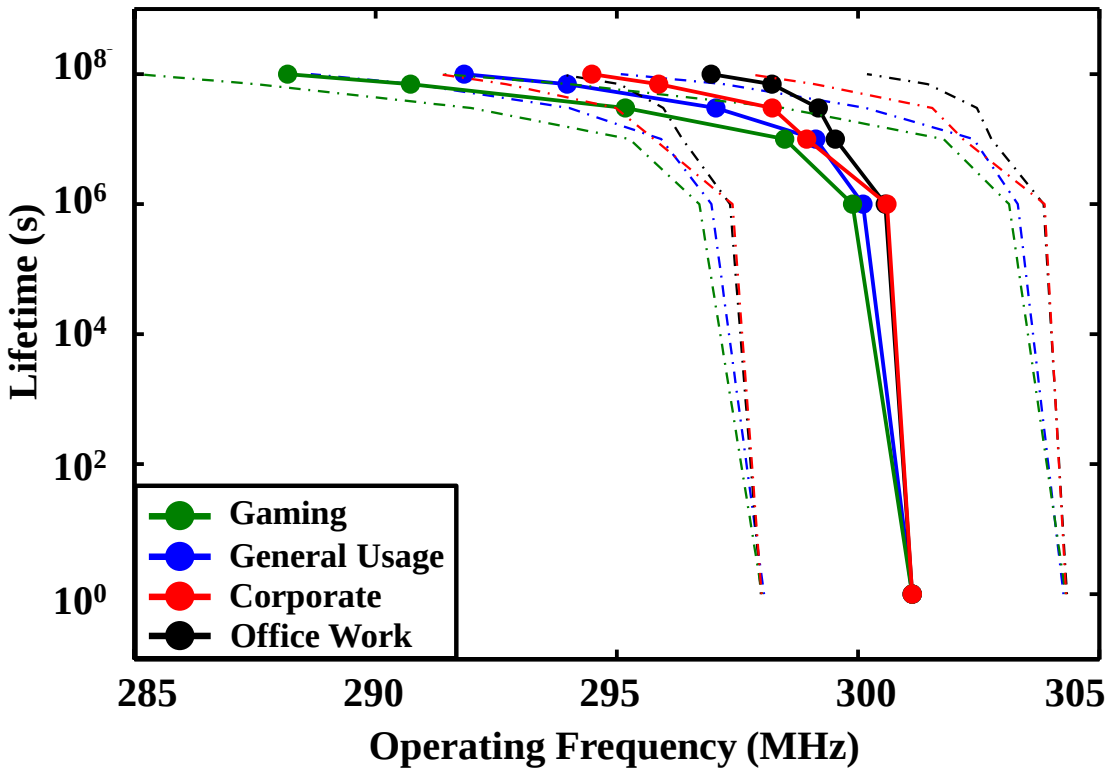


Figure 6.6: The estimated lifetimes of LEON3 microprocessor due to HCI for different use scenarios and different system frequencies. Dotted lines show the boundaries when considering process variation.

We have also estimated the lifetimes of the microprocessor under study based on different operating frequencies for different use scenarios due to GOBD, as shown in Figure 6.7. The results clearly indicate that the estimated system lifetimes decrease as the system frequency increases, and gaming has the shortest lifetime.

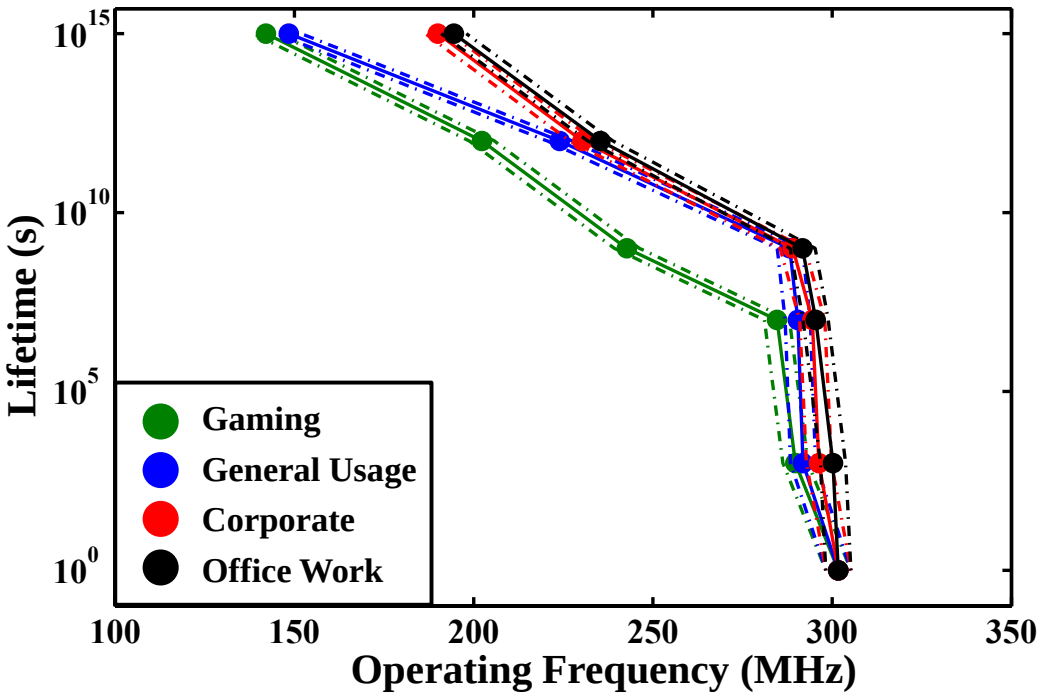


Figure 6.7: The estimated lifetimes of LEON3 microprocessor due to GOBD for different use scenarios and different system frequencies. Dotted lines show the boundaries when considering process variation.

6.1.2.2 Case Study 2: 32-bit RISC microprocessor

Besides LEON3, the 32-bit RISC microprocessor [123] which includes around 73k gates was also analyzed and studied.

Figures 6.8, 6.9 and 6.10 show the estimated lifetime due to BTI, HCI and GOBD, respectively, for the RISC microprocessor for different benchmarks based on different operating frequencies. Similar to the results for the LEON3, the results clearly indicate that the estimated system lifetimes decrease as the system frequency increases.

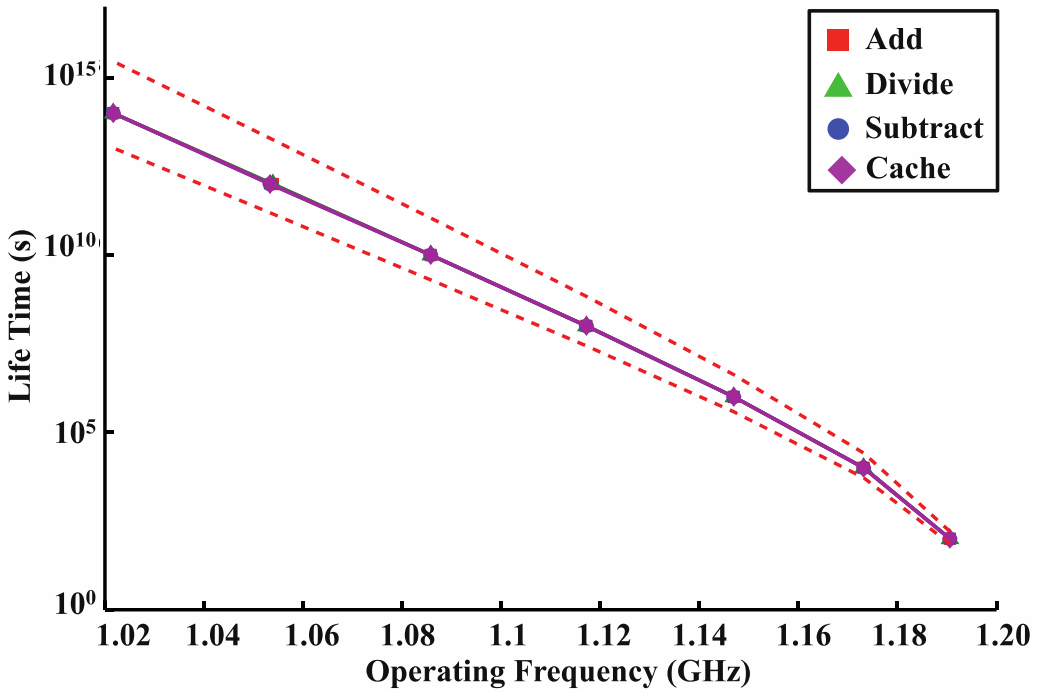


Figure 6.8: The estimated lifetimes of the RISC microprocessor due to BTI for different benchmarks and different system frequencies.

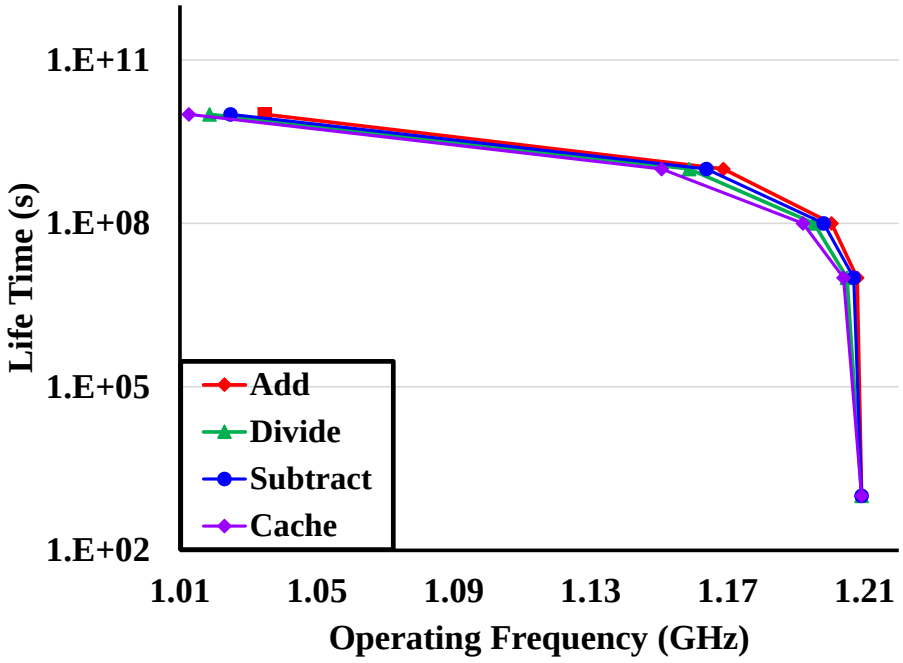


Figure 6.9: The estimated lifetimes of the RISC microprocessor due to HCI for different benchmarks and different system frequencies.

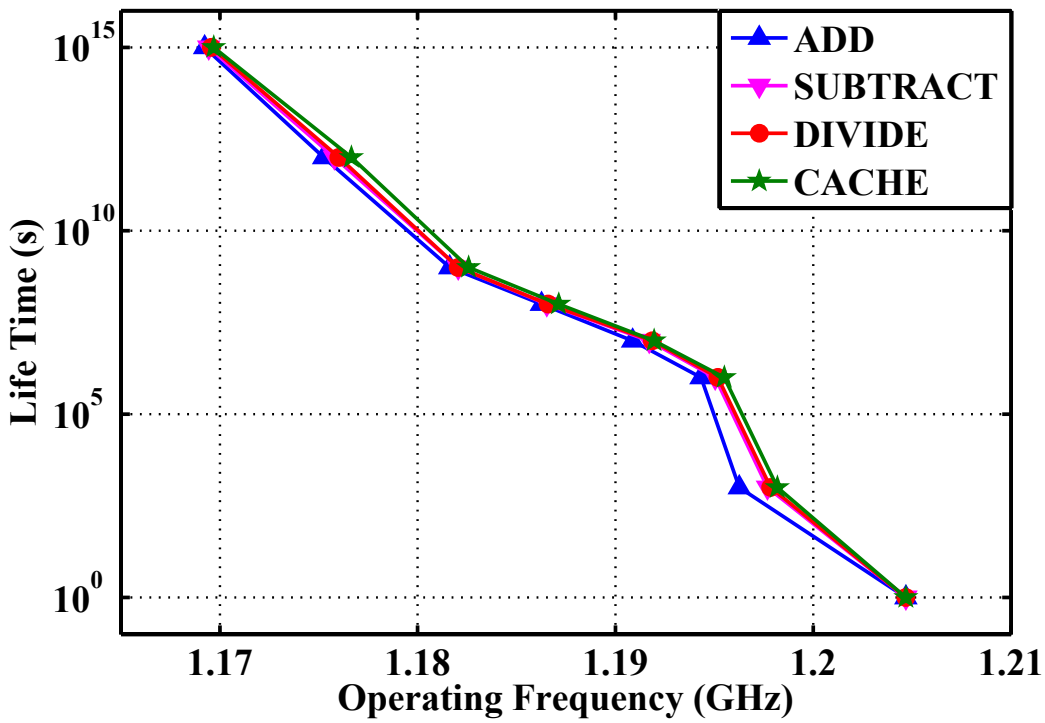


Figure 6.10: The estimated lifetimes of the RISC microprocessor due to GOBD for different benchmarks and different system frequencies.

6.2 Performance Degradation Analysis for Memory Blocks

6.2.1 SRAM Circuit

A 1024 word × 32 bit memory, which consists of 32,768 6T SRAM cells, shown in Figure 6.11, was generated with a memory generator [128] and was used for our study. Since each SRAM cell within the memory experiences different electrical stress and temperature when the microprocessor is executing benchmarks, each SRAM cell has a different threshold voltage drift due to BTI and HCI and different gate current leakage due to GOBD. The framework described in Section 4 was used to acquire the electrical and thermal profiles of the SRAM cells.

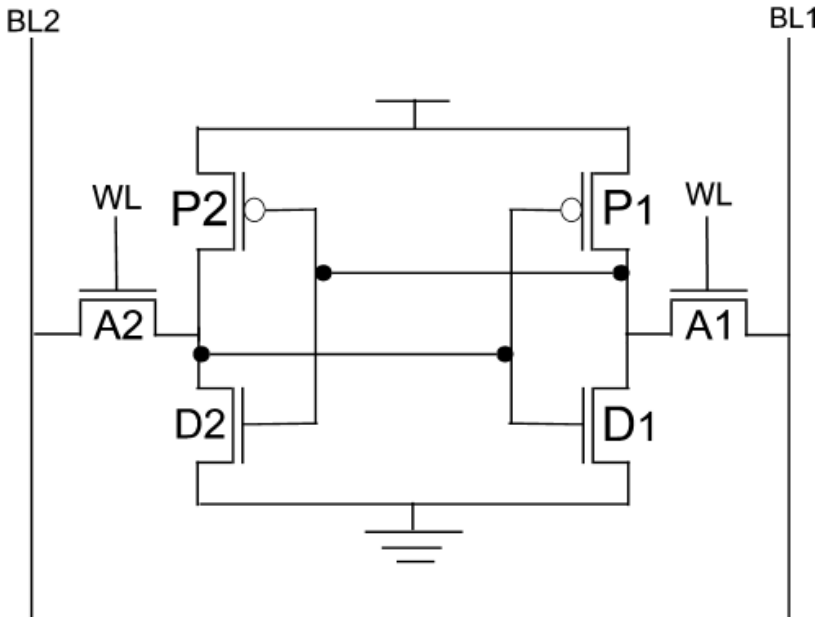


Figure 6.11: A typical 6T SRAM cell is shown.

As an SRAM cell undergoes AC and DC stress, the cell will become increasingly skewed as one device degrades more than the other. This leads to impaired noise immunity.

6.2.2 Memory Wearout Simulation Results

SRAMs are characterized with several performance metrics. These include the read and retention static noise margins (SNMs), write margin, read current (I_{READ}), and the minimum retention voltage (Vdd-min). The static noise margins are defined as the minimum DC noise voltage necessary to change the state of an SRAM cell. The read SNM is measured with the access transistors turned on, while the access transistors are off for the retention SNM. The write margin is the minimum voltage needed to flip the state of the cell, with the access transistors turned on. Vdd-min is the minimum voltage in which the SRAM retains its state. Finally, read current, which is inversely proportional to

access time, is the current flow through pull-down devices when performing a read operation.

A set of standard benchmarks were run on the microprocessor system under study. The electrical stress, as shown in Figures 6.12 and 6.13, and thermal profiles for the memory were collected by the aging assessment framework described in Section 4. The electrical and thermal profiles, together with the lifetime models from Section 3, were then used to analyze the performance of each SRAM cell within the memory.

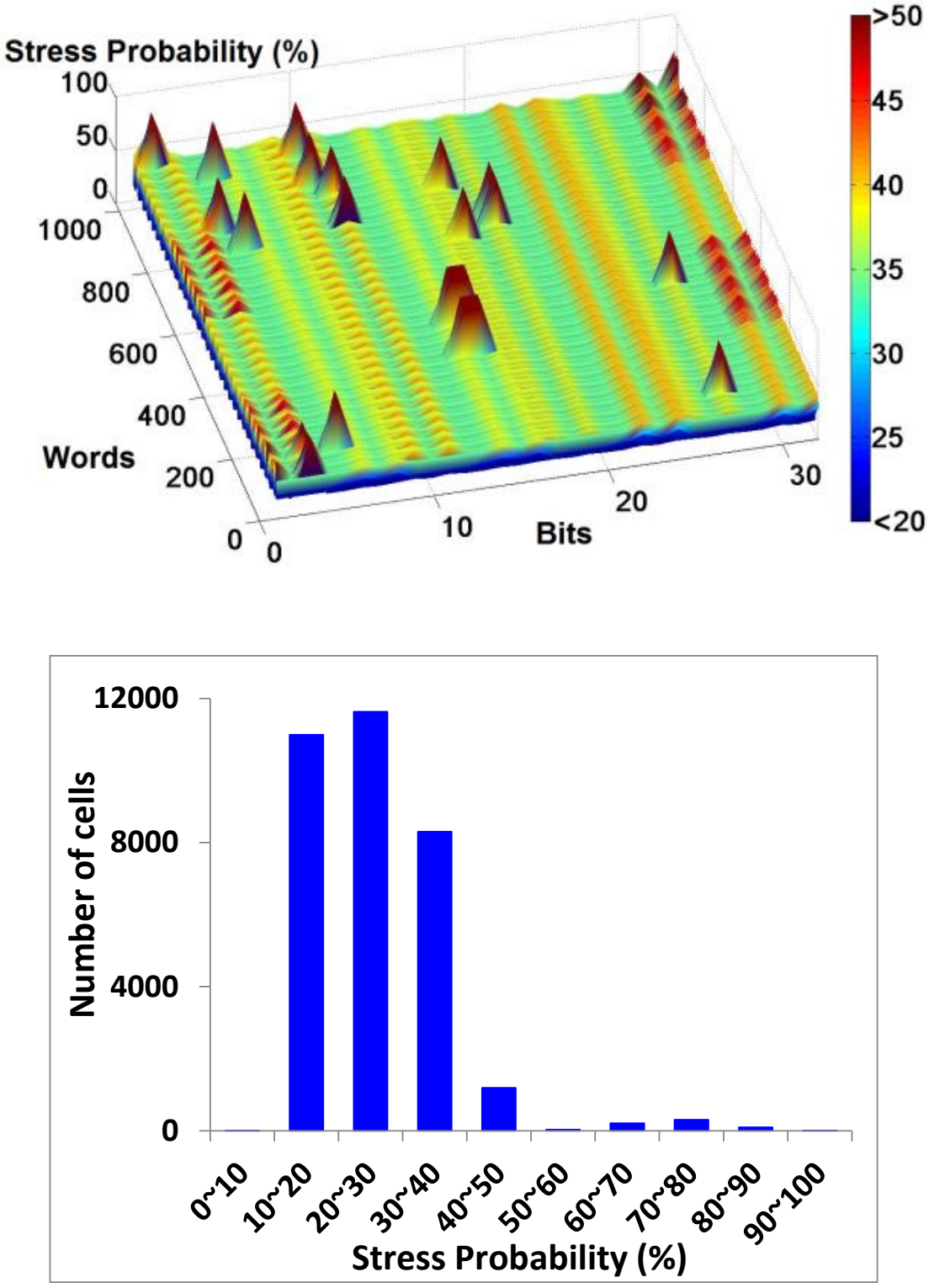


Figure 6.12: The distribution of the stress probability for the data cache while running a set of standard benchmarks.

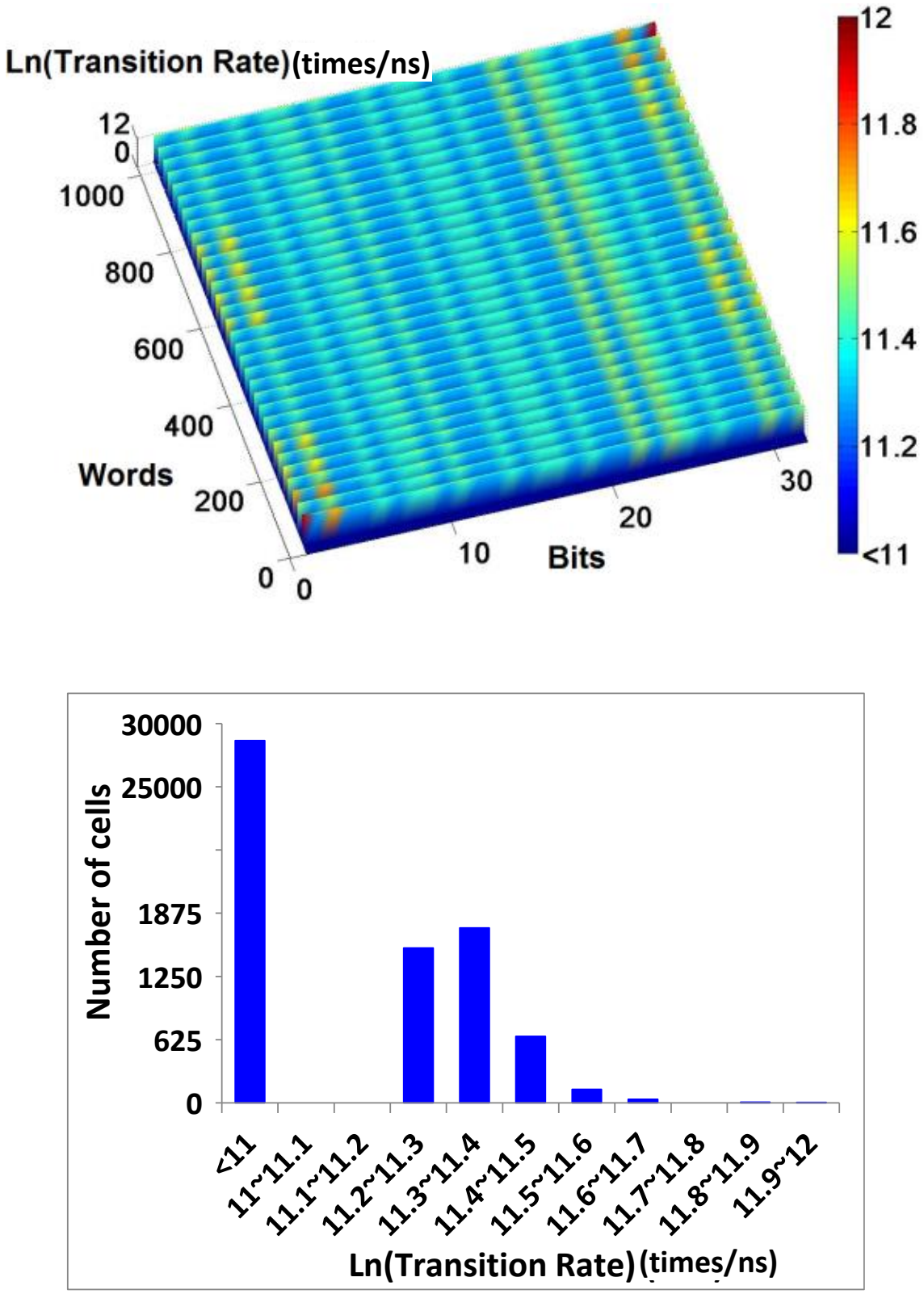


Figure 6.13: The distribution of the transition rate for the data cache while running a set of standard benchmarks.

Figures 6.14, 6.15 and 6.16 show the degradation of read SNM, write margin, read current, and Vdd-min of the memory due to BTI, HCI and GOBD for different use scenarios. For BTI and TDDB, the results show that the SRAM cells degrade differently while undergoing longer BTI stress. For HCI, the results show that some of the SRAM cells degrade and some of them improve while undergoing longer HCI stress.

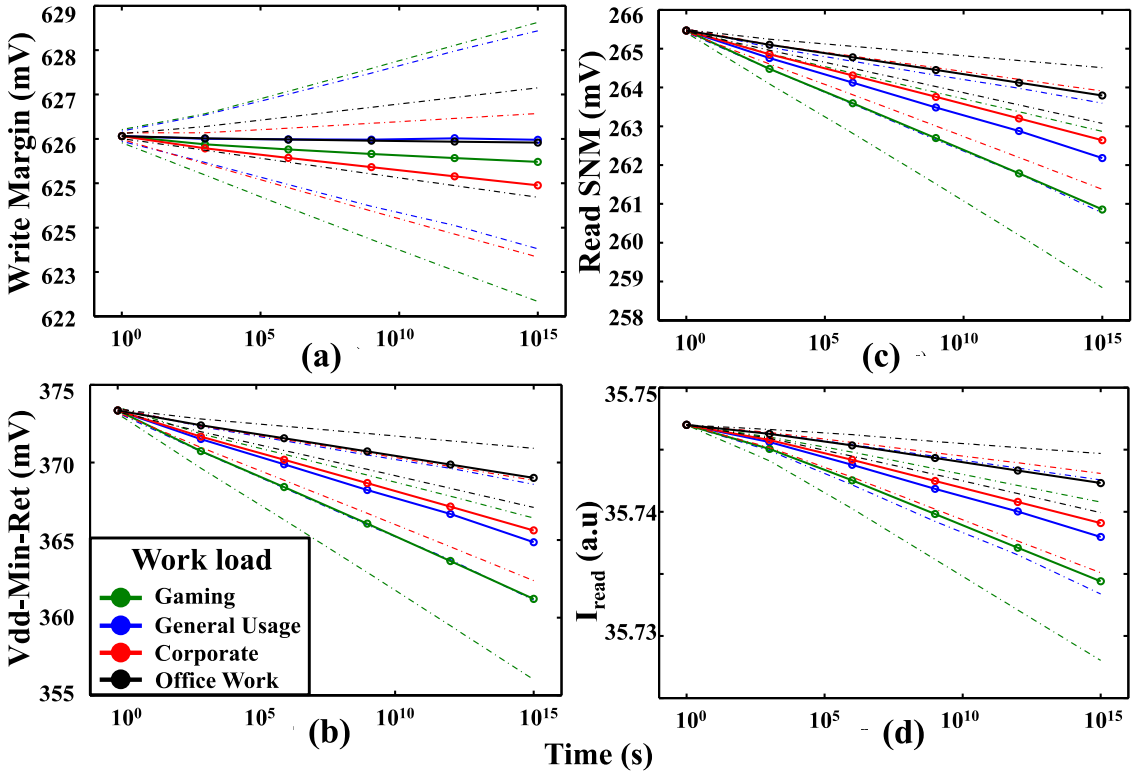


Figure 6.14: The degradation of (a) write margin, (b) vdd-min, (c) read SNM, and (d) read current of the memory due to BTI for different use scenarios.

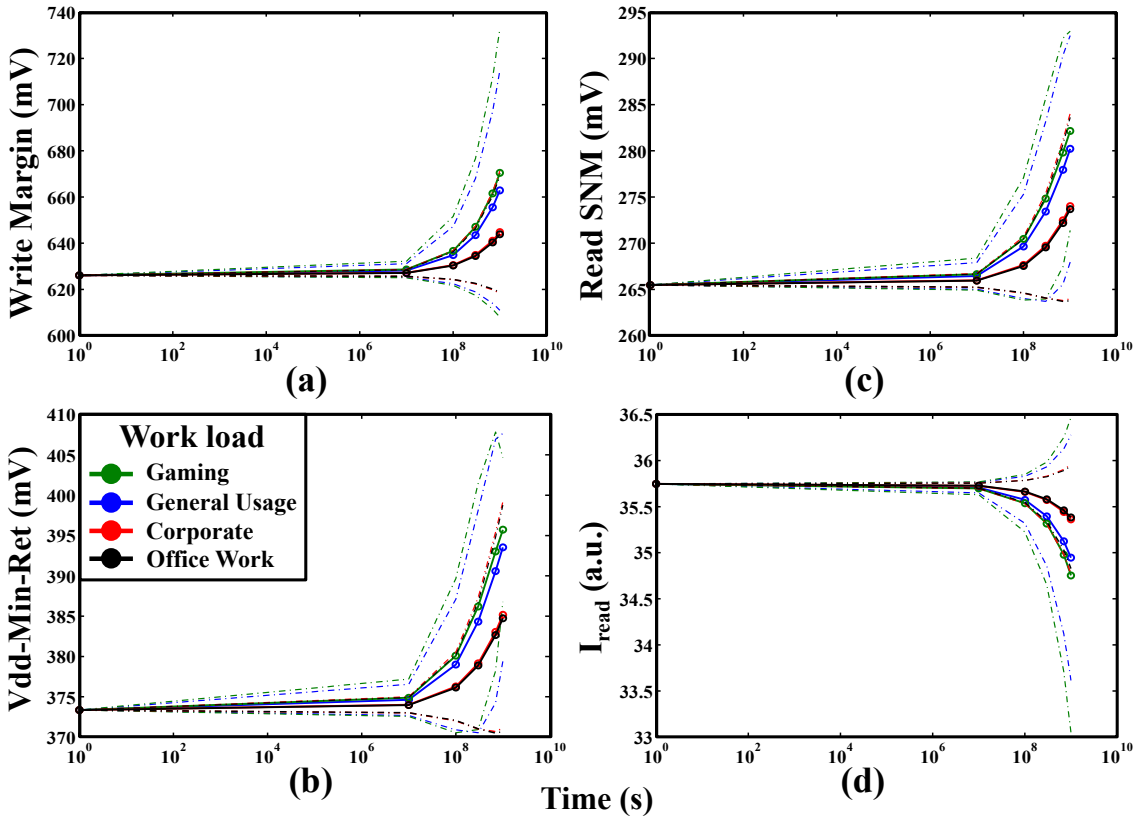


Figure 6.15: The degradation of (a) write margin, (b) vdd-min, (c) read SNM, and (d) read current of the memory due to HCI for different use scenarios.

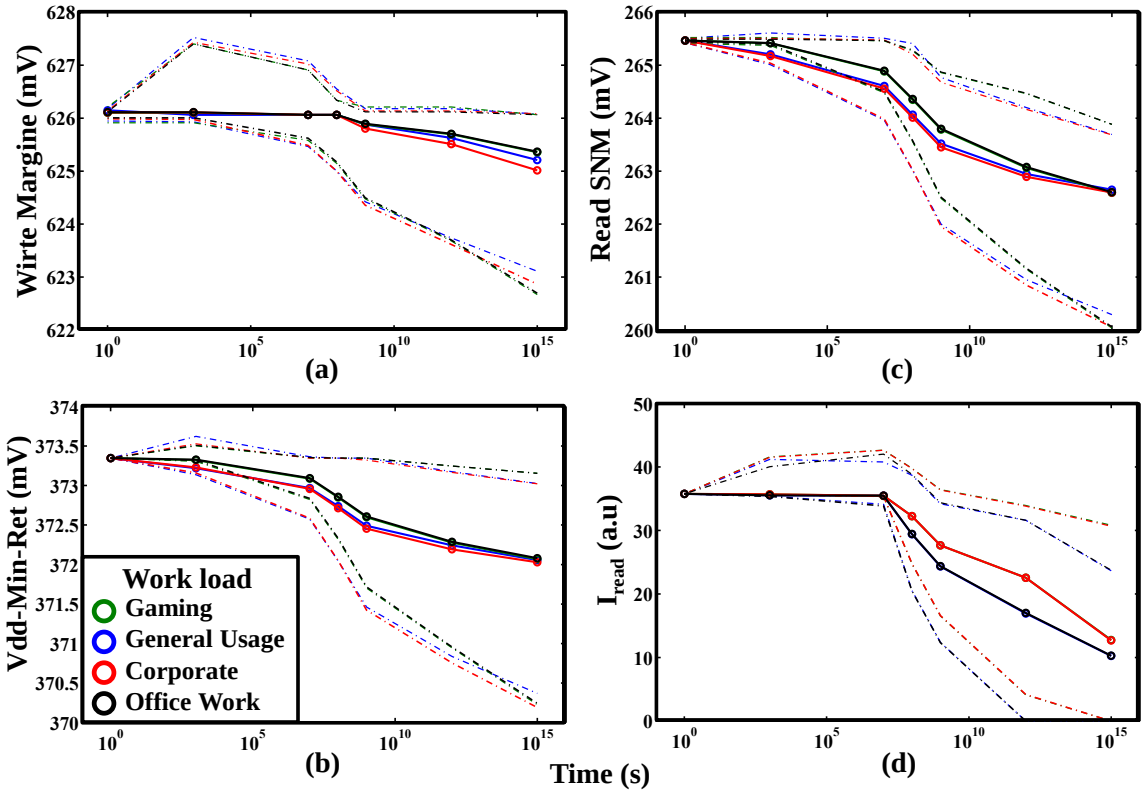


Figure 6.16: The degradation of (a) write margin, (b) vdd-min, (c) read SNM, and (d) read current of the memory due to GOBD for different use scenarios.

To better understand the impact of the frontend wearout mechanisms on the memory for different performance metrics, Figures 6.17, 6.18 and 6.19 show all performances under BTI, HCI and GOBD, respectively, normalized with respect to the specification and nominal fault free performance, i.e.,

$$Y = \left(X - X_{spec}\right) / \left(X_{fault-free} - X_{spec}\right), \tag{6.2}$$

where X is the performance parameter under study, $X_{fault\text{-}free}$ is the nominal performance, and X_{spec} is the design specification for fault-free operation.

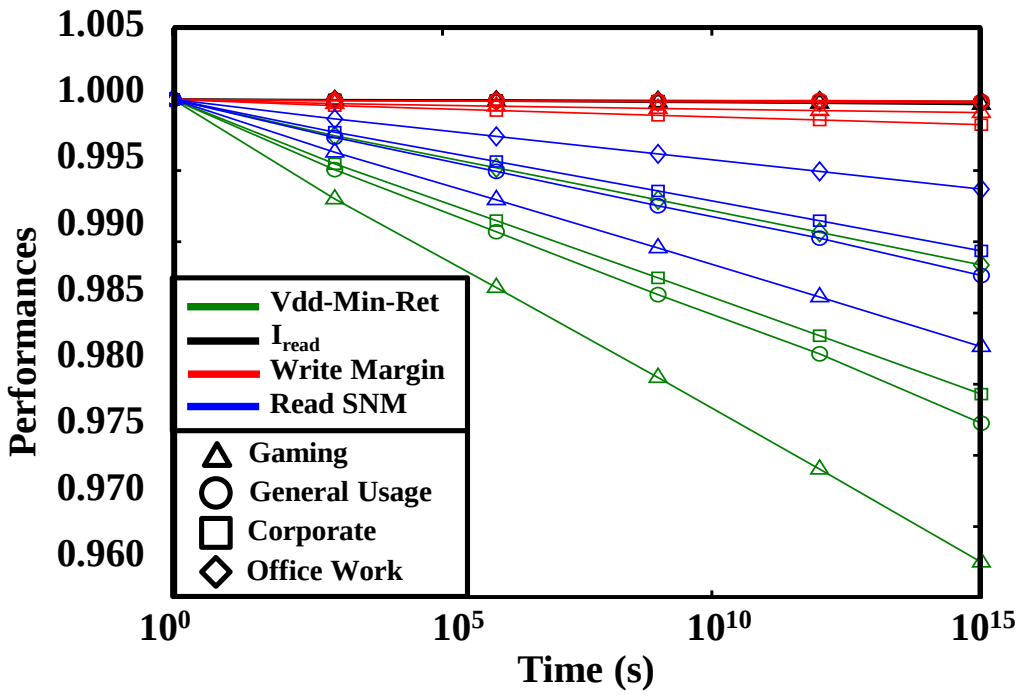


Figure 6.17: The performance metrics of the memory for different use scenarios under BTI.

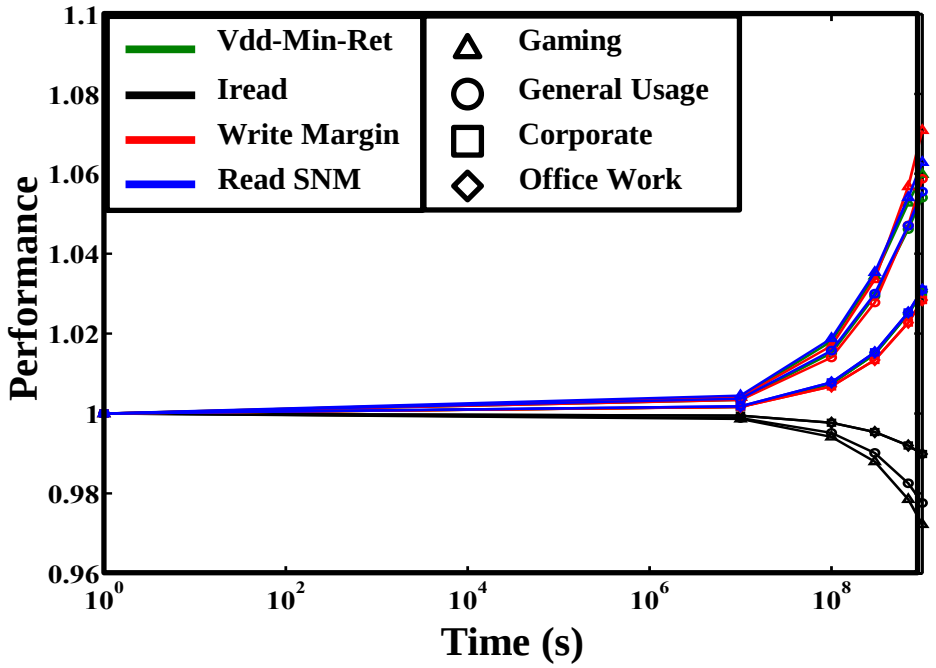


Figure 6.18: The performance metrics of the memory for different use scenarios under HCI.

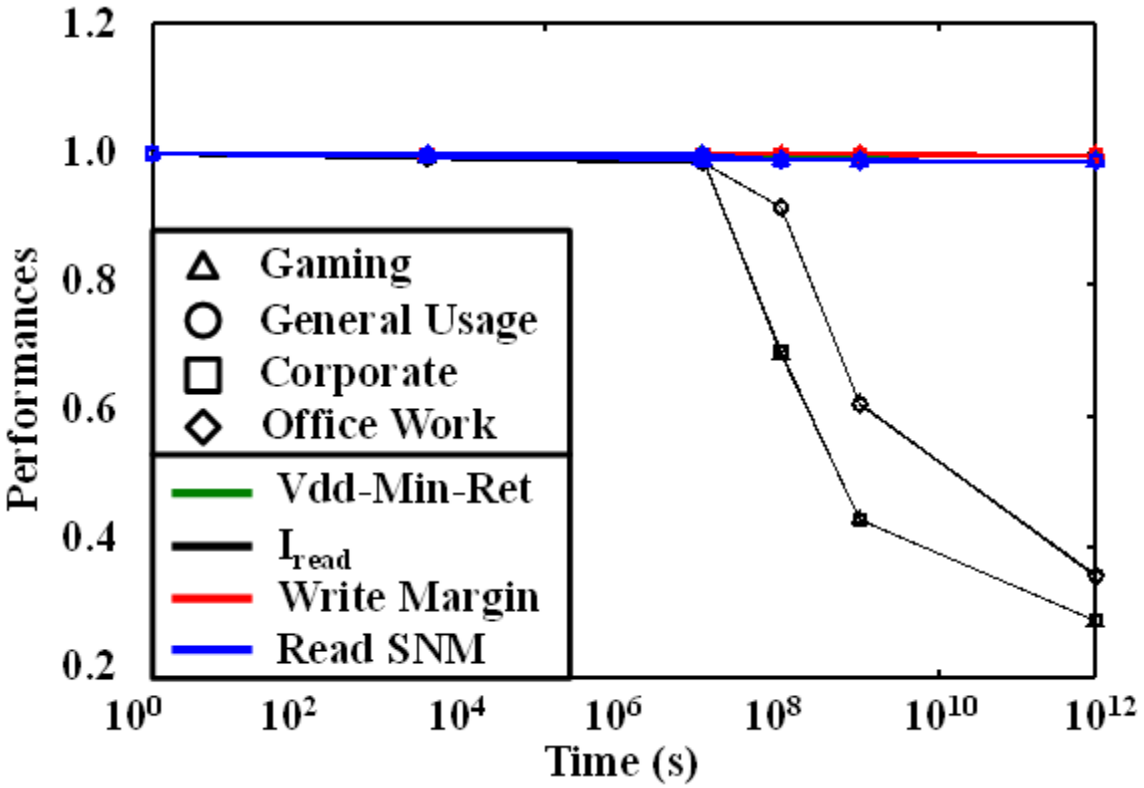


Figure 6.19: The performance metrics of the memory for different use scenarios under GOBD.

As our results indicate, for BTI, the minimum retention voltage is most strongly impacted, while the read stability is also severely affected. Both the write margin and read current are relatively unaffected by BTI. Regarding HCI, it improves the read SNM, write margin, and Vdd-min, and degrades I_{READ}. For GOBD, the read current is most severely impacted while the minimum retention voltage, read stability and write margin are relatively unaffected by GOBD.

CHAPTER 7

CONCLUSIONS

7.1 Conclusions of the Research

The research presents a simulation workflow to estimate lifetime for a variety of wearout mechanisms, including negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), gate oxide breakdown (GOBD), hot carrier injection (HCI), backend time dependent dielectric breakdown (BTDDB), electromigration (EM), and stress-induced voiding. Taking into account the detailed thermal and electrical stress profiles of microprocessor systems while running real-world applications, a methodology is developed to accurately assess microprocessor lifetime based on each wearout mechanism. In addition, this research presents a way to establish the link between the device-level wearout models, the electrical stress profile, the thermal profile, and system performances for both logic and memory blocks.

For BTDDB, the impact of line ends was studied and found to be clearly significant. These irregular geometries can potentially impact chip lifetime and need to be separately extracted and included in the reliability simulator.

The work identified the first block that is likely to fail in a system and takes into account a variety of use scenarios, composed of a fraction of time in operation, a fraction of time in standby, and a fraction of time when the system is off.

Since the memory blocks within the microprocessor were found to be more vulnerable than the other units, the research also provide a methodology to analyze memory performance degradation due to the frontend wearout mechanisms with studying

DC noise margins in conventional 6T SRAM cells as a function of NBTI, PBTI, HCI and GOBD degradation. This provides insights on memory reliability under realistic use conditions.

7.2 Future Work

While the microprocessor lifetime and performance degradation for each wearout mechanism have been analysed, the lifetime and performance degradation when considering all the wearout mechanisms impacting the system simultaneously hasn't been studied yet. For frontend wearout mechanisms, the impact of BTI and HCI on device threshold voltage and the impact of GOBD on device gate leakage can be further taken into account together. The system lifetimes due to frontend wearout mechanism can then be more realistically evaluated.

Similarly, the impact of BTDDB, EM and SIV on interconnects can be taken into account together. This will provide more realistic lifetime estimates for microprocessor systems.

APPENDIX A

LIFETIME WITH RECONFIGURATION THROUGH REDUNDANCY ALLOCATION

Error correcting codes can ensure that a memory system can tolerate faults. Our storage blocks include data/instruction cache which contain 1024 32-bit words, tag storages which contain 128 28-bit words, and register file which contains 256 32-bit words. BCH codes [129][130] require seven additional bits per word and can correct one bit per word.

To determine the impact of redundancy, let's first suppose that an SRAM cell, that stores one bit, is composed of I components, with Weibull parameters, $\eta_i, \beta_i, i = 1, \dots, I$. The survival rate of each cell depends on stress and temperature. Overall, the probability of survival of a cell is $R = exp\left(-\sum_i \left(\frac{TF}{\eta_i}\right)^{\beta_i}\right)$. If a word contains N bits (N=32 for our data cache), then in the absence of redundancy, the probability of survival of a word is $R_{word} = \prod_{j=1}^{N} R_j$. If there are M words (M=1024 for our data cache), the probability of survival of the memory is $R_{SRAM} = \prod_{k=1}^{M} R_{word,k}$. The characteristic lifetime is when only $e^{-1} = R_{SRAM}$ have survived. In this case, in the absence of redundancy,

$$1 = \sum_{k=1}^{M} \sum_{j=1}^{N} \sum_i \left(\frac{\eta_{SRAM}}{\eta_{ijk}}\right)^{\beta_{ijk}}. \quad \text{(A.1)}$$

If all cell experience the same stress profile, then

$$1 = MN \sum_i \left(\frac{\eta_{SRAM}}{\eta_i}\right)^{\beta_i}. \quad \text{(A.2)}$$

Now, let's suppose that the data cache uses error correcting codes for each memory word such that the word contains 39 bits and one bit can be corrected. The probability of survival of a word is $R_{word} = \prod_{j=1}^{N} R_j + \sum_{j=1}^{N}(1 - R_j)(\prod_{q \neq j}^{N} R_q)$. For our example, N=39. The probability of survival of the memory is $R_{SRAM} = \prod_{k=1}^{M} R_{word,k}$. Then, if $e^{-1} = R_{SRAM}$, corresponds to the characteristic lifetime, we solve

$$1 = -\sum_{k=1}^{M} ln\left(R_{word,k}(TF = \eta_{SRAM})\right) \quad \text{(A.3)}$$

to find the characteristic lifetime. If all cells experience the same stress profile, then $R_{word} = R^N + N(1 - R)R^{N-1}$ and the characteristic lifetime is the solution of

$$1 = -Mln(R^N + N(1 - R)R^{N-1}), \quad \text{(A.4)}$$

where R is evaluated at $TF = \eta_{SRAM}$.

REFERENCES

[1] C.-C. Chen, F. Ahmed, D. H. Kim, S. K. Lim, L. Milor, "Backend dielectric reliability simulator for microprocessor system," *Microelectronics Reliability*, vol. 52, pp. 1953-1959, 2012.

[2] C.-C. Chen, F. Ahmed, and L. Milor, "A Comparative Study of Wearout Mechanisms in State-of-Art Microprocessors," *Proc. IEEE International Conference on Computer Design*, pp. 271-276, 2012.

[3] C-C. Chen and L. Milor, "System-level modeling and microprocessor reliability analysis for backend wearout mechanisms," *Proc. Design Automation and Test in Europe*, pp. 1615-1620, 2013.

[4] C.-C. Chen, M. Bashir, L. Milor, D.H. Kim, and S.K. Lim, "Simulation of system backend dielectric reliability," *Microelectronics Journal*, 2014.

[5] 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.

[6] C.-C. Chen, F. Ahmed, L. Milor, "Impact of NBTI/PBTIon SRAMs within microprocessor systems: Modeling, simulation, and analysis," *Microelectronics Reliability*, vol. 53, pp. 1183-1188, 2013.

[7] C.-C. Chen, S. Cha, T. Liu, and L. Milor, "System-level modeling of microprocessor reliability degradation due to BTI and HCI," *IEEE International Reliability Physics Symposium*, pp. CA.8.1-CA.8.9, 2014.

[8] C.-C. Chen, S. Cha and L. Milor, "System-level modeling of microprocessor reliability degradation due to GOBD," *Conference on Design of Circuits and Integrated Systems*, 2014.

[9] C.-C. Chen, M. Bashir, L. Milor, D.H. Kim, and S.K. Lim, "Backend dielectric chip reliability simulator for complex interconnect geometries," *IEEE International Reliability Physics Symposium*, pp. BD.4.1-BD.4.8, 2012.

[10] M. Bashir, C.-C. Chen, L. Milor, D.H. Kim, and S.K. Lim, "Backend dielectric reliability full chip simulator," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 8, pp. 1750-1762, Auguest 2014.

[11] R. Kwasnick, A.E. Papathanasiou, M. Reilly, A. Rashid, B. Zaknoon, and J. Falk, "Determination of CPU use conditions," *Proc. Int. Reliability Physics Symp.*, 2011, pp. 2C.3.1-2C.3.6.

[12] W. Wang, S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu, and Y. Cao, "The impact of NBTI effect on combinational circuit: Modeling, simulation, and analysis," *IEEE Trans. VLSI*, vol. 18, no. 2, pp. 173-183, Feb. 2010.

[13] J.B. Velamala, K.B. Sutaria, V.S. Ravi, and Y. Cao, "Failure analysis of asymmetric aging under NBTI," *IEEE Trans. Device & Material Reliablity*, vol. 13, no. 2, pp. 340-349, June 2013.

[14] R. Zheng, J. Velamala, V. Balakrishnan, E. Mintarno, S. Mitra, S.Krishnan, and Y. Cao, "Circuit aging prediction for low-power operation," *Proc. Custom Integrated Circuits Conf.,* 2009, pp. 427-430.

[15] S. Gupta and S.S. Sapatnekar, "GNOMO: Greater-than-NOMinal Vdd operation for BTI mitigation," *Proc. Asia & South Pacific Design Automation Conf.,* 2012, pp. 271-276.

[16] F. Firouzi, S. Kiamehr, and M.B. Tahoori, "Statistical analysis of BTI in the presence of process-induced voltage and temperature variations," *Proc. Asia & South Pacific Design Automation Conf.,* 2013, pp. 594-600.

[17] S. Kiamehr, F. Firouzi, and M.B. Tahoori, "Aging-aware timing analysis considering combined effects of NBTI and PBTI," *Proc. Int. Symp. Quality Electronic Design,* 2013, pp. 53-59.

[18] G. La Rosa, W.L. Ng, S. Rauch, R. Wong, and J. Sudijono, "Impact of NBTI induced statistical variation to SRAM cell stability," *Proc. Int. Reliability Physics Symp..,* 2006, pp. 274-282.

[19] S.V. Kumar, C.H. Kim and S.S. Sapatnekar, "Impact of NBTI on SRAM read stability and design for reliability," *Proc. Int. Symp. Quality Electronic Design,* 2006, pp. 213-218.

[20] 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. Int. Reliability Physics Symp.,* 2012, pp. 2F.2.1-2F.2.5.

[21] J. Fang and S.S. Sapatnekar, "The impact of BTI variations on timing in digital logic circuits," *IEEE Trans. Device & Materials Reliability*, vol. 13, no. 1, pp. 277-286, March 2013.

[22] V. Huard, C. Parthasarathy, C. Guerin, T. Valentin, E. Pion, M. Mammasse, N. Planes, and L. Camus, "NBTI degradation: From transistor to SRAM arrays," *Proc. Int. Reliability Physics Symp.,* 2008, pp. 289-300.

[23] M. Choudhury, V. Chandra, K. Mohanram, and R. Aitken., "Analytical model for TDDB-based performance degradation in combinational logic." in *Design, Automation & Test in Europe Conf.,* 2010, pp. 423-428.

[24] M.H. Woods, "MOS VLSI reliability and yield trends," *Proc. of the IEEE*, vol. 74, no. 12, pp. 1715-1729, 1986.

[25] C.T. Rosenmayer, et al., "Effect of stresses on electromigration," *Proc. Int. Reliability Physics Symp.*, 1991, pp. 52-56.

[26] S.P. Hau-Riege and C.V. Thompson, "Experimental characterization and modeling of the reliability of interconnect trees," *J. Applied Physics*, pp. 601-609, Jan. 2001.

[27] A. Zitzelsberger, et al., "On the use of highly accelerated electromigration test (SWEAT) on copper," *Proc. Int. Reliability Physics Symp.*, 2003, pp. 161-165.

[28] D-Y. Kim and S.S. Wong, "Mechanism for early failure in Cu dual damascene structure," *Proc. Int. Interconnect Technology Conf.*, 2003, pp. 265-267.

[29] J.S. Suehle, "Ultrathin gate oxide reliability: physical models, statistics, and characterization," *IEEE Trans. Electron Devices*, vol. 49, no. 6, pp. 958-971, June 2002.

[30] M. Depas, T. Nigam, and M.M. Heyns, "Soft breakdown of ultra-thin gate oxide layers," *IEEE Trans. Electron Devices*, vol. 43, no. 9, pp. 1499-1504, Sept. 1996.

[31] C. Hu and Q. Lu, "A unified gate oxide reliability model," Proc. Int. Reliability Physics Symp., 1999, pp. 47-51.

[32] D. DiMaria, "Hole trapping, substrate currents, and breakdown in thin silicon dioxide films," *IEEE Trans. Electron Device Letters*, vol. 16, no. 5, pp. 184-186, May 1995.

[33] R. Degraeve, *et al.*, "New insights in the relation between electron trap generation and the physical properties of oxide breakdown," *IEEE Trans. Electron Devices*, vol. 45, no. 4, pp. 904-910, April 1998.

[34] S. Cha, C.-C. Chen, and L. Milor, "Frontend wearout modeling and parameter extraction for circuits with I/O measurments," *IEEE International Integrated Reliability Workshop*, October 2014.

[35] S. Cha, W. Kim, and L. Milor, "Gate Oxide Breakdown Parameter Extraction with Ground and Power Supply Signature Measurements," *Conference on Design of Circuits and Integrated Systems*, 2014.

[36] C. Hu, *et al.*, "Hot-electron-induced MOSFET degradation – model, monitor, and improvement," *IEEE Trans. Electron Devices*, vol. 32, no. 2, pp. 375-384, Feb. 1985.

[37] W. Weber, "Dynamic stress experiments for understanding hot-carrier degradation phenomena," *IEEE Trans. Electron Devices*, vol. 35, no. 9, pp. 1476-1486, Sept. 1988.

[38] W. Weber, *et al.*, "Hot-carrier degradation of CMOS-inverters," *Proc. Int. Electron Devices Meeting*, 1988, pp. 208-211.

[39] K.N. Quader, et al., "A bidirectional NMOSFET current reduction model for simulation of hot-carrier-induced circuit degradation," *IEEE Trans. Electron Devices*,

vol. 40, no. 12, pp. 2245-2254, Dec. 1993.

[40] X. Li, et al., "Compact modeling of MOSFET wearout mechanisms for circuit-reliability simulation," *IEEE Trans. on Device and Materials Reliability*, vol. 8, pp. 98-121, 2008.

[41] S. Zafar *et al.,* "A comparative study of NBTI and PBTI (charge trapping) in S_iO_2/HFO_2 stacks with FUSI, TiN, Re Gates," *Proc. Symp. VLSI Tech.,* 2006, pp. 23-25.

[42] M. Denais, *et al*., "Interface trap generation and hole trapping under NBTI and PBTI in advanced CMOS technology with a 2nm gate oxide," *IEEE Trans. Device and Materials Reliability*, vol. 4, pp. 715-722, Dec. 2004.

[43] M. Alam and S. Mahapatra, "A comprehensive model of PMOS NBTI degradation," *Microelectronics Reliability*, vol. 45, pp. 71-81, 2005.

[44] C. Yu, *et al*., "Impact of temperature-accelerated voltage stress on PMOS RF performance," *IEEE Trans. Device and Materials Reliability*, vol. 4, no. 4, pp. 664-669, Dec. 2004.

[45] S. Zafar, et al., "A model for negative bias temperature instability (NBTI) in oxide and high k pFETs," *Proc. Symp. on VLSI Technology*, 2004, pp. 208-209.

[46] M. Bashir and L. Milor, "Modeling low-k dielectric breakdown to determine lifetime requirements," *IEEE Design and Test of Computers*, vol. 26, no. 6, pp. 18-27, 2009.

[47] M. Bashir and L. Milor, "Analysis of the impact of linewidth variation on low-k dielectric breakdown," *Proc. Int. Reliability Physics Symp.,* 2010, pp. 895-902.

[48] L. Milor and C. Hong, "Area scaling for backend dielectric breakdown," *IEEE Trans. Semiconductor Manufacturing*, vol. 23, no. 2, pp. 429-441, 2010.

[49] Z.C. Wu, *et al.,* "Leakage mechanism in Cu damascene structure with methylsilane-doped low-k CVD oxide as intermetal dielectric," *IEEE Electron Device Letters*, vol. 22, no. 6, pp. 263-265, June 2001.

[50] R. Gonella, P. Motte, and J. Torres, "Time-dependent-dielectric breakdown used to assess copper contamination impact on inter-level dielectric reliability," *Int. Reliability Workshop Final Report*, 2000, pp. 189-190.

[51] J. Noguchi, *et al.,* "TDDB improvement in Cu metallization under bias stress," *Proc. Int. Reliability Physics Symp.,* 2000, pp. 339-343.

[52] C. Hong and L. Milor, "Porosity-induced electric field enhancement and its impact on charge transport in porous inter-meal dielectrics," *Proc. Int. Reliability Physics Symp.,* 2006, pp. 588-589.

[53] C. Hong and L. Milor, “Effect of porosity on charge transport in porous ultra-low-k dielectrics,” *Proc. Int. Interconnect Technology Conf.*, 2006, pp. 140-142.

[54] E.T. Ogawa, *et al.,* “Stress-induced voiding under vias connected to wide Cu metal leads,” *Proc. Int. Reliability Physics Symp.,* 2002, pp. 312-321.

[55] K. Doong, *et al.,* “Stress-induced voiding and its geometry dependency characterization,” *Proc. Int. Reliability Physics Symp.*, 2003, pp. 156-160.

[56] C.J. Zhai, *et al.,* “Simulation and experiments of stress migration for Cu/low-k BeoL,” *IEEE Trans. Device and Materials Reliablity*, vol. 4, no. 3, pp. 523-529, Sept. 2004.

[57] J. Srinivasan, *et al.,* “Lifetime reliability: Towards an architectural solution,” *IEEE Micro*,vol. 25, pp. 70-80, 2005.

[58] E. Karl, *et al.,* “Multi-mechanism reliability modeling and management in dynamic systems,” *IEEE Trans. VLSI,* vol. 16, no. 4, pp. 476-487, April 2008.

[59] J. Srinivasan, *et al.*, “The Case for Lifetime Reliability-Aware Microprocessors,” *Proc. ISCA*, pages 276–287, 2004.

[60] J. Shin *et al.*, “A Framework for Architecture-Level Lifetime Reliability Modeling,” *Proc. DSN*, pages 534–53, 2007.

[61] M. Bashir, *et al.,* “Methodology to determine the impact of linewidth variation on chip scale copper/low-k backend dielectric breakdown,” *Microelectronics and Reliability*, vol. 50, nos. 9-11, pp. 1341-1346, Sept.-Nov. 2010.

[62] M. Bashir, *et al.,* “Backend low-k TDDB chip reliability simulator,” *Proc. Int. Reliability Physics Symp.,* 2011, pp. 65-74.

[63] M. Bashir and L. Milor, "Towards a chip level reliability simulator for copper/low-k backend processes," *Proc. DATE*, 2010, pp. 279-282.

[64] G. La Rosa, *et al.*, “Impact of NBTI induced statistical variation to SRAM cell stability,” *Proc. IRPS.,* 2006, pp. 274-282.

[65] S.E. Rauch, “Review and reexamination of reliability effects related to NBTI-induced statistical variations,” *IEEE Trans. Device & Materials Reliability,* vol. 7, no. 4, pp. 524-530, Dec. 2007.

[66] S.V. Kumar, C.H. Kim and S.S. Sapatnekar, “Impact of NBTI on SRAM read stability and design for reliability,” *Proc. ISQED,* 2006, pp. 213-218.

[67] F. Ahmed and L. Milor, “Reliable cache design with on-chip monitoring of NBTI degradation in SRAM cells using BIST,” *Proc. VLSI Test Symp.*, 2010, pp. 63-68.

[68] F. Ahmed and L. Milor, "NBTI resistant SRAM design," *Proc. Int. Workshop on Advances in Sensors and Interfaces*, 2011, pp. 82-87.

[69] R. Kanj, *et al.*, "An elegant hardware-corroborated statistical repair and test methodology for conquering aging effects," *Proc. ICCAD*, 2009, pp. 497-504.

[70] V. Huard, *et al.*, "NBTI degradation: From transistor to SRAM arrays," *Proc. IRPS*, 2008, pp. 289-300.

[71] Z, Chishti, *et al.*, "Improving cache lifetime reliability at ultra-low voltages," *Proc. Micro*, 2009, pp. 89-99.

[72] N.S. Kim, *et al.*, "Analyzing the impact of joint optimizatoin of cell size, redundancy, and ECC on low-voltage SRAM array tool area," *IEEE Trans. VLSI*, vol. 20, no. 12, pp. 2333-2337, Dec. 2012.

[73] Y. Xiang, *et al*., "System-level reliability modeling for MPSoCs," *Proc. CODES+ISSS*, 2010, pp. 297-306.

[74] J. Srinivasan, *et al*., "Lifetime reliability: Toward an architectural solution," *IEEE Micro*, vol. 25, pp. 70-80, 2005.

[75] E. Karl, *et al*., "Multi-mechanism reliability modeling and management in dynamic system," *IEEE Trans. VLSI*, vol. 16, no. 4, pp. 476-487, Apr. 2008.

[76] A. Strong, E. Wu, R. Vollertsen et al., "Reliability Wearout Mechanisms in Advanced CMOS Technologies," *Wiley-IEEE Press*, 2009.

[77] G. S. Haase and J. W. McPherson, "Modeling of interconnect dielectric lifetime under stress conditions and new extrapolation methodologies for time-dependent dielectric breakdown," *Proc IRPS*, 2007, pp. 390-398.

[78] F. Chen, et al., "A comprehensive study of low-k SiCOH TDDB phenomena and its reliability lifetime model development," *Proc. IRPS*, 2006, pp. 46-53.

[79] K-Y. Yiang, H.W. Yao, and A. Marathe, "TDDB Kinetics and their Relationship with the E- and √E-models", *Proc. Interconnect Technology Conf.*, 2008, pp. 168-170.

[80] F. Lacopi, Z. Tokei, M. Stucchi, F. Lanckmans, and K. Maex, "Diffusion barrier integrity and electrical performance of Cu/porous dielectric damascene lines," *IEEE Electron Device Lett*., vol. 24, no. 3, pp. 147-149, Mar. 2003.

[81] V.L. Lo, K.L. Pey, C.H. Tung and D.S. Ang, "A critical voltage triggering irreversible gate dielectric degradation", *Proc. IEEE International Reliability Physics Symposium*, 2007, pp.576-577.

[82] R. W. I. de Boer, N. N. Iosad, A. F. Stassen, T. M. Klapwijk, and A. F. Morpurgo, "Influence of the gate leakage current on the stability of organic single-crystal field-effect transistors," *Appl. Phys. Lett.*, vol. 86, no. 3, p. 032 103, Jan. 2005.

[83] T.K. Wong, "Time Dependent Dielectric Breakdown in Copper Low-k Interconnects: Mechanisms and Reliability Models". *Materials*, vol. 5, pp. 1602-1625, 2012.

[84] Mibench benchmark: http://www.eecs.umich.edu/mibench

[85] E. T. Ogawa, K.-D. Lee, V. A. Blaschke, and P. S. Ho, "Electromigration reliability issues in dual-damascene Cu interconnections," *IEEE Trans. Reliab.*, vol. 51, pp. 403–419, 2002.

[86] A. S. Oates, F. Nlkansah, and S. Chittipeddi, "Electromigration-induced drift failure of via contacts in multilevel metallization," *J. Appl. Phys.*, vol. 72, no. 6, pp. 2227–2231, Sep. 1992.

[87] Z.-S. Choi et al., "Activation energy and prefactor for surface electromigration and void drift in Cu interconnects", *J. Applied Physics*, vol. 102, pp. 083509 - 083509-4, 2007.

[88] R. L. de Orio et al., "A compact model for early electromigration lifetime estimation", *Proc. SISPAD*, 2011, pp. 23-26.

[89] A. S. Oates et al., "Electromigration Failure Distributions of Cu/Low-k Dual-Damascene Vias: Impact of the Critical Current Density and a New Reliability Extrapolation Methodology", *IEEE Trans. Device Materials Reliability*, vol. 9, pp. 244-254, 2009.

[90] R. R. Morusupalli et al., "Comparison of Line stress predictions with measured electromigration failure times", *IEEE Int. Integrated Reliability Workshop Final Report*, 2007, pp. 124-127.

[91] SoC Encounter: www.cadence.com/products/di/soc_encounter/pages/default.aspx

[92] H. W. Yao et al., "Stress migration model for Cu interconnect reliability analysis", *J. Applied Physics*, vol. 110, pp. 073504 - 073504-5, 2011.

[93] C. M. Tan et al., "Stress migration model for stress-induced voiding in integrated circuit interconnections," *Applied Physics Letters*, vol. 91, pp. 061904 - 061904-3, 2007.

[94] S. Pae, et al., "BTI reliability of 45 nm high-K + metal-gate process technology," *IEEE International Reliability Physics Symposium.*, 2008, pp. 352-357.

[95] 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*, vol. 58, no. 8, pp. 2743-2751, 2011.

[96] S. Cha, C.-C. Chen and L. Milor, "System-level estimation of threshold voltage degradation due to NBTI with I/O measurements," *IEEE International Reliability Physics Symposium*, pp. PR.1.1-PR.1.7, 2014.

[97] 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 VLSI Test Symposium (VTS)*, pp. 1-6, 2014.

[98] 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.

[99] 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 SiO_2/HFO_2 stacks with FUSI, TiN, Re Gates," *Proc. Symp. VLSI Tech.*, 2006, pp. 23-25.

[100] F. Ahmed and L. Milor, "NBTI resistant SRAM design," in *4th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*, 2011, pp. 82-87.

[101] S.-Y. Chen, C.-H. Tu, P.-W. Kao, M.-H. Lin, H.-S. Haung, J.-C. Lin, M.-C. Wang, S.-H. Wu, Z.-W. Jhou, S. Chou, and J. Ko, "Investigation of DC hot-carrier degradation at elevated temperatures for p-channel metal-oxide-semiconductor field-effect transistors," *Jpn. J. Appl. Phys*, vol. 47, no. 3, pp. 1527-1531, 2008.

[102] W. Wang, V. Reddy, A.T. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, "Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology," *IEEE Trans. Device and Materials Reliability*, vol. 7, no. 4, pp. 509-517, 2007.

[103] C. Ma, B. Li, L. Zhang, J. He, X. Zhang, X. Lin, and M. Chan, "A Unified FinFET Reliability Model Including High K Gate Stack Dynamic Threshold Voltage, Hot Carrier Injection, and Negative Bias Temperature Instability," *Proc. Int. Symp. Quality Electronic Design*, 2009, pp. 7-12.

[104] C.H. Tu, S.Y. Chen, A.E. Chuang, H.S. Huang, Z.W. Jhou, C.J. Chang, S. Chou, and J. Ko, "Transistor variability after CHC and NBTI stress in 90 nm pMOSFET technology," *Electronics Letters*, vol. 45, no. 15, pp. 854-856, 2009.

[105] K. Okada, "Analysis of the relationship between defect site generation and dielectric breakdown utilizing A-mode stress induced leakage current." *IEEE Trans. on Electron Devices*, vol. 47, no. 6, pp. 1225-1230, 2000.

[106] R. Degraeve, et al. "A new model for the field dependence of intrinsic and extrinsic time-dependent dielectric breakdown." *IEEE Trans. Electron Devices*, vol. 45, no. 2, pp. 472-481, 1998.

[107] S. Takagi, N. Yasuda, and A. Toriumi, “Experimental evidence of in- elastic tunneling and new I–V model for stress-induced leakage current,” *IEDM Tech. Dig.*, 1996., pp. 323-326.

[108] R. Degraeve, B. Kaczer, A. De Keersgieter, and G. Groeseneken. "Relation between breakdown mode and breakdown location in short channel NMOSFETs and its impact on reliability specifications." *IEEE Int. Reliability Physics Symp.*, 2001, pp. 360-366.

[109] B.P. Linder, D.J. Frank, J.H. Stathis, and S.A. Cohen, “Transistor-lim- ited constant voltage stress of gate dielectrics,” in *Symp. VLSI Technol. Dig.*, 2001, pp. 93–94.

[110] B. E. Weir *et al.*, “Gate oxide reliability projection to the sub-2 nm regime,” *Semicond. Sci. Technol.*, vol. 5, pp. 455–461, 2000.

[111] E. Y. Wu, E. J. Nowak, A. Vayshenker, W. L. Lai, and D. L. Harmon, “CMOS scaling beyond the 100-nm node with silicon-dioxide-based gate dielectrics,” *IBM Journal of Research and Development*, vol. 46, 2002.

[112] T. Nigam, A. Kerber, and P. Peumans. "Accurate model for time-dependent dielectric breakdown of high-k metal gate stacks." *IEEE Int. Reliability Physics Symposium*, 2009, pp. 523-530

[113] 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." *IEEE Int. Reliability Physics Symposium*, 2013.

[114] J.H. Stathis, “Percolation models for gate oxide breakdown,” *J. Appl. Phys.*, vol. 86, no. 10, pp. 5757–5766, Nov. 1999.

[115] E. Miranda and J. Suñé. "Analytic modeling of leakage current through multiple breakdown paths in SiO2 films." *IEEE Int. Reliability Physics Symp.* 2001, pp. 367-379.

[116] Degraeve, Robin, et al. "New insights in the relation between electron trap generation and the statistical properties of oxide breakdown." *IEEE Trans. Electron Devices*, vol. 45, no. 4 pp. 904-911, 1998.

[117] Xilinx ISE: http://www.xilinx.com/products/design-tools/ise-design-suite

[118] PrimeTime power modeling tool: www.synopsys.com

[119] HotSpot: www.lava.cs.virginia.edu/HotSpot

[120] L. Milor and C. Hong, “Area scaling for backend dielectric breakdown,” *IEEE Trans.Semiconductor Manufacturing*, vol. 23, no. 3,m pp. 429-441, Aug. 2010.

[121] W.R. Hunger, “The analysis of oxide reliability data,” *Proc. Int. Integr. Rel. Workshop Final Rep.*, 1998, pp. 114-134.

[122] LEON3 processor: www.gaisler.com

[123] J. Draper, *et al.,* **"**Implementation of a 32-bit RISC processor for the data-intensive architecture processing-in-memory chip", *Proc. of the IEEE International Conference on Application-Specific Systems, Architectures and Processors*, 2002, pp.163 – 172, 17-19 July 2002.

[124] Launchbird Design Systems Inc. *CF FFT*. http://www.opencores.org.

[125] NCSU EDA. *NCSU Free PDK45*. http://www.eda.ncsu.edu/wiki/FreePDK.

[126] Design Compiler: http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DesignCompiler/Pages/default.aspx

[127] SoC Encounter RTL-to-GDSII System: http://www.cadence.com/products/di/soc_encounter/pages/default.aspx

[128] Memory compiler: www.arm.com

[129] A. Hocquenghem, "Codes Correcteurs D'erreurs," *Chiffres*, vol. 2, pp.147-156, Sept. 1959.

[130] R.C. Bose and D.K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Information and Control*, vol. 3, pp. 68-79, Mar. 1960.