HotAging — Impact of Power Dissipation on Hardware Degradation

Frank Sill Torres[1,2] Alberto Garcia Ortiz[3] Rolf Drechsler [1,2]

[1]Cyber-Physical Systems, DFKI GmbH, Bremen, Germany
[2]Group of Computer Architecture, University of Bremen, Germany
[3]Institute of Electrodynamics and Microelectronics, University of Bremen, Germany

***Abstract*—Safety and dependability are of utmost importance for many integrated systems. Hence, it must be guaranteed throughout the whole system's lifetime that no ambient and internal influences can affect the system's integrity. Under this scope and having in mind the side-effects of today's nanoscale technologies, hardware degradation is of rising concern. However, related studies should not solely focus on aging effect itself, but also consider its relation to any accelerating factors, especially temperature. Towards this end, this work presents a study on how the power dissipation of a circuit, and thus, its temperature, can expedite wear-out effects. Therefore, three different analysis are performed—aging without and with consideration of temperature and the study on how guard-banding strategies are affected. In order to distinguish random and, maliciously intended or accidentally produced, worst case scenarios, we implemented an algorithm that determines a combination of input vectors that forces high aging states and high power dissipation. Results indicate that aging under consideration of temperature can increase circuit delay by more than 26% (random case) and by nearly 40% (worst case). That means, if a maximum acceptable delay degradation is defined, designs can enter malfunction states already in a period of weeks (worst case) or months (random case). These results underline the importance of considering power dissipation, and thus temperature, when doing aging analysis and aging verification.**

***Index Terms*—Aging, Temperature, BTI, Degradation, Power**

I. Introduction

Safety is an essential requirement for almost any electronic system and must be guaranteed throughout its whole lifetime. Amongst the several threats that deteriorate safety, wear-out mechanisms are assuming a continuously increasing role. This is mainly motivated by two basic tendencies—technology scaling and performance demands. While the former leads to more intense shaping of aging mechanisms, the later reduces acceptable consequences of degradation.

Designers have access to a wide collection of countermeasures, including guard-banding, i. e. the artificial extension of the critical timing path [1], runtime testing [2] or concurrent error detection schemes like transition detection [3] and spatial redundancy [4]. Furthermore, several analysis environments for modeling of aging impact on integrated circuits have been developed. This includes, but is not limited to, reliability aware circuit simulators [5], [6] or aging-dependent cell libraries [7]. In can be noted, though, that the developed approaches focus mainly on the data dependency of wear-out mechanisms but discard the impact of circuit temperature on aging. This seems to be careless, having in mind the exponential impact of temperature on hardware degradation. Even more, the relation between aging and temperature can be applied in order to enhance malicious attacks to the hardware as reported by [8] or [9]. The fundamental idea of these attacks is the extraction of input vectors that lead to highest degradation and the application of these vectors during runtime. Simulation results indicate that typical processors could be set into failure state in less than a year.

Circuit temperature is directly related to power dissipation, and thus, to switching activity–a relation that is explored by several available tools that determine the on-chip temperature. Having additionally in mind the data-dependency of today's most significant wear-out mechanisms, like *Bias Temperature Instability* (BTI), *Time-Dependent Dielectric Breakdown* (TDDB) or *Hot Carrier Injection* (HCI), one can expect that each circuit has a set of input vectors that lead to disastrous combination of strong wear-out conditions and high power dissipation.

The goal of this work is the study on how power dissipation, and consequently temperature, impacts circuit aging. Towards this end, we generated a flow that enables the determination of data-dependent aging due to BTI, the dominating wear-out mechanism in today's technologies [7], [10], and the acceleration of this degradation by high temperatures. Furthermore, we provide an evolutionary approach for determining the worst case for temperature accelerated aging.

This remainder of this work is organized as follows. Section II introduces the implemented analysis flow, Section III discusses the results and Section IV draws the conclusions.

II. Analysis Environment

This section describes the implemented environment that enables the exploration of the relation between BTI-based aging and temperature in integrated circuits.

A. Overview

The complete flow is depicted in Fig. 1. Its inputs are the design netlist and an aging-aware cell library against which the design has been synthesized. Both are processed by the implemented *HotAging* algorithm that generates random and worst case input vectors in terms of degradation and power dissipation. Next, a sub-flow for delay and power analysis estimates the power profiles for the given input vectors and the maximum delays if aging is considered. The tool HotSpot [11] is applied for extracting the circuit temperatures for using the determined power profiles. Finally, the acceleration of design wear-out due to temperatures and the impact on guard-banding is calculated.

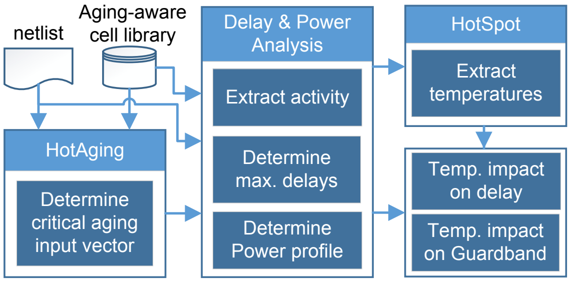


Fig. 1: Complete analysis flow

Algorithm 1: Hot-Aging

```
   Input: Netlist $N = (C, W, I, O)$, cell library $L$
   Output: Pair of input vectors $IP$
1  Initialize(N,L)
   // Detemine worst aging input vector
2  while max generations without improvements do
3      $AuxParent \leftarrow Parent$
4      for $i \leftarrow 1$ to $\lambda$ do
5          $AuxOffspr \leftarrow$ Mutation($P_m/2, Parent$)
6          EvaluateFitnessAging($AuxOffspr, N$)
7          ReplaceIfBetter($AuxParent, AuxOffspr$)
8      end
9      $Parent \leftarrow AuxParent$
10 end
   // Detemine worst hot-aging vector
11 while max generations without improvements do
12     $AuxParent \leftarrow Parent$
13     for $i \leftarrow 1$ to $\lambda$ do
14         $AuxOffspr \leftarrow$ Mutation($P_m, Parent$)
15         EvaluateFitnessAging($AuxOffspr, N$)
16         EvaluateFitnessPower($AuxOffspr, N$)
17         ReplaceIfBetter($AuxParent, AuxOffspr$)
18     end
19     $Parent \leftarrow AuxParent$
20 end
21 Return $Parent$
```

B. Aging-aware cell library

Aging analysis on circuit level offers high accuracy and is recommended for analog or small digital circuits. Unfortunately, its very poor scalability prevents the application for more complex designs. Hence, we opted for the generation of an aging-aware cell library, as presented in [7]. The library is based on the open-source design kit FreePDK45 that applies a high-performance 45nm Predictive Technology Model (PTM) [12]. The actual cell characterization has been done with the tool Cadence Liberate.

Cell aging has been modeled as BTI, having in mind its dominating impact on hardware degradation in current technologies [10]. As extensively discussed in [8], [9], [13], aging due to BTI depends on stress and recovery periods, i. e. the time that certain voltage levels have been applied to the transistor terminals. For a standard cell, these times are directly related to the duty-cycle of the cell's inputs [7], [8]. Therefore, we characterized for each cell different versions that differ in the duty-cycle of the inputs. The related technology parameters for each degradation level are based on the data reported in [13].

In detail, we defined the maximum impact of BTI-aging on threshold voltages of pMOS and nMOS devices over a 10 years period operated at a temperature of 125 °C with $\Delta v_{th,p}$=0.16 V and $v_{th,n}$=0.11 V . The maximum degradation of the mobilities μ_n and μ_p of both device types over 10 years operated at 125 °C was defined with $\Delta\mu_p$=$-0.08\mu_p$ and $\Delta\mu_n$=$-0.07\mu_n$.

The impact of the duty-cycle on the BTI-aging of a transistor is modeled using the analytical models proposed in [13]. Therefore, we generated for both device types 11 modified transistor models, with v_{th} and μ varying in steps of 10 % duty-cycle. During cell library characterization, these transistor models have been used to characterize the cells for different duty-cycles of each input. Thus, each 2-input cell has 121 versions. In order to keep the library size manageable, we opted against cells with higher input number and implemented a library with 11 cells that differ in implemented logic and drive strength.

C. HotAging

The goal of the proposed flow is the exploration of the critical combination of high power dissipation, and thus, high circuit temperatures, and unfavorable signal states that put devices into high degradation conditions. In order to fulfill both requirements, we implemented a two-part solution starting with the determination of critical aging-aware input vectors and followed by the extraction of input vectors that lead to high switching activity.

Having in mind the immense search-space, we opted for an evolutionary algorithm (EA), listed in Algorithm 1. The proposed solution adopts a direct encoding in which the n inputs are represented by a genome consisting of $2 \cdot n$ binary numbers which mean the values of two consecutive input vectors. In accordance with the literature, e. g. [14], [15], the mutation probability P_m=$1/(2n)$ has been chosen in all experiments. The literature offers a large selection of various recombination operators for evolutionary algorithms, e. g. in [14], [16]. But since recombination could not yield any performance advantage in the present task, none of these operators has been used here. Finally, we chose a configuration of σ=1 parent and $lambda$=6 offspring, which have been indicated to yield a high sequential efficiency [15], [17].

The input values of the implemented EA are a cell library L and a design netlist N, consisting of a set C of cells, a set W of wires and two sets I and O containing the design inputs and outputs. During initialization (line 1), the cell arcs of all critical paths are determined, using the netlist and the cell library. A cell arc represents an input-to-output pair of a cell and the corresponding delay [18]. Furthermore, the parent genome is initialized and reset, i. e. the initial input vectors are all 0.

Next, the input vector that would lead to maximum delay increase due to BTI-aging is determined. Therefore, a generation of λ=6 offspring is created from the parent genome, whereas both consecutive vectors have the same values. For each offspring, the fitness in terms of aging is determined (line 6). During this analysis, it is verified for all cell timing arcs that are part of a critical path whether the input has value 0. This input value puts the connected pMOS devices into high BTI-aging state [13]. We opted against a consideration of aging in nMOS devices, i. e. forcing input value 1, due to the considerably higher impact of BTI on pMOS in the chosen technology [7]. The related fitness results from the relation between the cell arcs with input 0 and all cell arcs located in critical paths. Finally, the fitness of all offspring and the current parent are compared and the individual with the highest fitness is chosen as new parent (line 7). This procedure is repeated until the maximum number of generations without improvements, i. e. no higher fitness, is reached.

Starting from this resulting vector, the input vector with the best compromise between high BTI-aging and high switching activity is determined. Therefore, a generation of λ=6 offspring is generated from the parent genome, whereas both consecutive input vectors can have differing values. This is followed by the fitness evaluation with emphasis on aging (line 15) as described above. During this analysis, it must be considered that the second input vector differs from the first one. Next, the fitness in terms of switching power dissipation is extracted. This is done via the determination of the switched capacitance if both input vectors are applied consecutively (line 16). The fitness results form the relation between the switched capacitance and the sum of all capacitances in the design. Finally, the products of the fitness related to aging and power dissipation of all offspring and the current parent are compared and the individual with the highest value is chosen as new parent (line 17). This procedure is again repeated until the maximum number of generations without improvements is reached. The last parent is then chosen as final result.

Additionally, the algorithm generates sets of random input vectors (not shown) that are applied for comparison purposes throughout the analysis flow.

D. Delay and Power Analysis

The determination of the maximum delays and the power profiles starts with the extraction of the internal activity and the duty-cycles of each cell input using Modelsim and the input vectors generated by the *HotAging* algorithm. Next, the appropriate duty-cycle dependent cell type, provided by the aging-aware library, is allocated to each cell. Finally, the power profiles as well as the maximum design delay before and after aging are extracted with Synopsys DesignCompiler.

E. Temperature Consideration

The tool HotSpot (version 6.0) [11] is applied for the determination of the circuit temperature for the power profiles determined in the previous step. The tool is configured with standard parameters, using a reference temperature of 27 °C and design dependent values for heat sink and spreader dimensions.

The impact of temperature on BTI-aging is modeled with the widely applied acceleration factor $AF_{BTI}^{T_{op},T_0}$ listed in Eq. 1 [19]:

$$AF_{BTI}^{T_{op},T_0} = \exp\left[\frac{E_{aBTI}\cdot(T_{op}-T_0)}{k_B\cdot T_{op}\cdot T_0}\right] \tag{1}$$

with E_{aBTI} is the activation energy, which has been reported to vary from 0.1 eV to 0.84 eV [20] and which was set to E_{aBTI}=0.58 as recommended in [13]. Furthermore, T_{op} and T_0 represent the operation and the reference temperature, whereas the latter is set to 125 °C following from the degradation models applied for cell library characterization. Finally, k_B means the Boltzmann's constant.

In order to determine the circuit delay if operated at temperature T_{op} we extracted a model for the degradation of the threshold voltage v_{th} and the mobility μ. The model approximates the curves as $a\cdot t^b$, with a and b are the respective model parameters[1] and t is the time in seconds.

The dependence of the design delay t_d on Δv_{th} and μ can be roughly approximated with following equation [21]:

$$t_d \propto \left[\mu^{-2}\left(V_{DD}-v_{th}-\Delta v_{th}\right)^2\right]^{-1} \tag{2}$$

By using this equation, the acceleration factor, the parameters of the applied 45nm technology and the approximated Δv_{th} and $\Delta\mu$ models we generated a new model that predicts the delay degradation due to BTI $\Delta t_{d,T_{op}}$ in a period of t_{10}=10 years if operated at the temperature T_{op}.

$$\Delta t_{d,T_{op}} = \frac{\Delta t_{d,T_0}}{0.42\cdot t_{d,init}}\left(t_{10}\cdot AF_{BTI}^{T_{op},T_0}\cdot 3E10^{-1}\right)^{5.2567^{-1}} \tag{3}$$

with $\Delta t_{d,T_0}$ means the degradation of the design delay if operated at the reference temperature T_0 and $t_{d,init}$ represents the pristine delay.

F. Guard-band consideration

The application of a guard-band, i.e. the provision of an extra-delay to the delay of the critical path, is a common approach in order to circumvent the impact of parameter variations as well as aging [1]. In order to determine the time until a certain guard-band is missed, i.e. the delay of a path is above the initial delay plus the guard-band, we generated a model that predicts the circuit age t_{age} in seconds at which the BTI degradation leads to a delay increase Δt_{guard} if the circuit is operated at the operating temperature T_{op}.

$$t_{age} = 3E10\cdot\left(AF_{BTI}^{T_{op},T_0}\right)^{-1}\left(\Delta t_{guard}\frac{0.42\cdot t_{d,init}}{\Delta t_{d,T_0}}\right)^{5.2567} \tag{4}$$

III. Analysis Results

This section presents and discusses the obtained results of the executed experiments.

A. Motivation

In order to explore the relation between power and aging, this work performs three different analysis. Firstly, we compare the aging of random inputs with that of the developed *HotAging* algorithm, assuming a constant temperature independent of the power dissipation. This analysis shows the effect of worst case inputs for aging without considering temperature, as investigated in [8], [9]. Afterwards we compare the aging of random inputs and that of the *HotAging* algorithm considering the real temperature on the chip. The results quantify the importance of considering power dissipation, and thus temperature, when doing aging analysis and aging verification. Finally, we evaluate how the common anti-aging method guard-banding is affected if power is considered.

B. Setup

We implemented the environment presented in II and executed an extensive analysis using the EPFL benchmark suite [22].

The obtained results are listed in Tab. I. Initially, we extracted the delay $t_{d,init}$ of the pristine versions of the circuits (see column "Pristine delay") and the delay increase within ten years if the circuits are operated at 125 °C. Next, we determined the delay increase without consideration of temperature, i.e. all circuits are operated at 27 °C, for randomized input vectors (column "Delay/Rdm.") and input vectors extracted from the *HotAging* algorithm presented in section II (column "Delay/H.A."). Please note that the results for randomized input vectors are the average values of 10 different simulations with randomized input vectors, whereas the standard deviations was always below 10%. Furthermore, we determined the impact of high temperatures on the circuit degradation for randomized input vectors and vectors determined with the *HotAging* algorithm. We applied the input vectors extracted by the *HotAging* algorithm with a period equal to the circuit delay. In contrast, the input period of the randomized circuits was estimated for each input randomly, with the circuit delay as lower bound. Additionally, we simulated 20 copies of each circuit placed side by side. The extracted temperatures and the resulting acceleration factors, calculated with Eq. 1, are listed for randomized input vectors in columns "Temperature/Rdm." and "Accel. Factor/Rdm.", while its counterparts for input vectors extracted with the *HotAging* algorithm are located in columns "Temperature/H.A." and "Accel. Factor/H.A.". Column "Delay @ T_{op}" lists the circuit delay if the circuits are operated at the corresponding operation temperatures reported columns "Temperature/Rdm." and "Temperature/H.A.". Finally, columns "T.t. miss 15% guard" and "T.t. miss 10% guard" show the time until the circuit misses the maximum allowed delay if guard-bands of 15% or 10% are assumed and circuit temperature is considered.

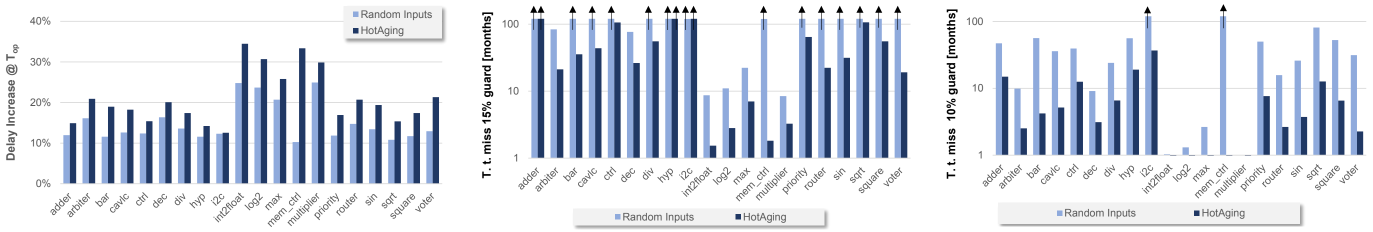
C. Discussion

Fig. 3 depicts the percentage increase of the circuit delays due to BTI-aging without consideration of temperature. The results indicate that randomized input vectors lead to an average delay increase of 12% with values up to 16% (dec).

[1] Following model parameters have been extracted for the chosen technology: $\Delta v_{th,pMOS}$: a=1.41E-3, b=125.1E-3, $\Delta v_{th,nMOS}$: a=2.8E-3, b=180.1E-3, μ_{pMOS}: a=1.0166, b=-4E-3, $\Delta\mu_{pMOS}$: a=1.008, b=-4E-3

TABLE I: Simulation Results

Benchmark		Delay @ 27 °C			Temperature T_{op}		Accel. Factor		Delay @ T_{op}		Time to guard 15%		Time to guard 10%	
Name	#Cells	Pristine	Rdm.	H.A.	Rdm.	H.A.	Rdm.	H.A.	Rdm.	H.A.	Rdm.	H.A.	Rdm.	H.A.
adder	1509	9.87E-09 s	1.10E-08 s	1.13E-08 s	27.6 °C	28.7 °C	4.2E-03	4.5E-03	1.11E-08 s	1.13E-08 s	2.9E+5 h	9.1E+4 h	3.4E+4 h	1.1E+4 h
arbiter	4847	1.51E-09 s	1.75E-09 s	1.79E-09 s	30.0 °C	36.1 °C	4.9E-03	7.7E-03	1.76E-09 s	1.83E-09 s	6.1E+4 h	1.5E+4 h	7.2E+3 h	1.8E+3 h
bar	2804	6.17E-10 s	6.77E-10 s	6.92E-10 s	38.5 °C	60.8 °C	9.1E-03	3.8E-02	6.88E-10 s	7.33E-10 s	3.4E+5 h	2.6E+4 h	4.1E+4 h	3.1E+3 h
cavlc	557	8.92E-10 s	9.99E-10 s	1.03E-09 s	30.1 °C	37.9 °C	5.0E-03	8.7E-03	1.00E-09 s	1.05E-09 s	2.2E+5 h	3.2E+4 h	2.6E+4 h	3.8E+3 h
ctrl	111	3.24E-10 s	3.62E-10 s	3.66E-10 s	31.1 °C	40.1 °C	5.4E-03	1.0E-02	3.64E-10 s	3.74E-10 s	2.4E+5 h	7.7E+4 h	2.9E+4 h	9.1E+3 h
dec	309	2.80E-10 s	3.23E-10 s	3.28E-10 s	30.5 °C	37.8 °C	5.1E-03	8.7E-03	3.26E-10 s	3.36E-10 s	5.6E+4 h	1.9E+4 h	6.6E+3 h	2.3E+3 h
div	35949	1.83E-07 s	2.07E-07 s	2.13E-07 s	28.9 °C	30.5 °C	4.6E-03	5.1E-03	2.07E-07 s	2.14E-07 s	1.5E+5 h	4.0E+4 h	1.7E+4 h	4.8E+3 h
hyp	203240	6.70E-07 s	7.46E-07 s	7.62E-07 s	28.5 °C	29.4 °C	4.4E-03	4.7E-03	7.47E-07 s	7.65E-07 s	3.4E+5 h	1.2E+5 h	4.1E+4 h	1.4E+4 h
i2c	868	6.23E-10 s	6.96E-10 s	6.91E-10 s	29.8 °C	35.9 °C	4.9E-03	7.6E-03	6.99E-10 s	7.01E-10 s	8.6E+4 h	2.2E+5 h	8.6E+4 h	2.7E+4 h
int2float	155	5.84E-10 s	6.68E-10 s	6.87E-10 s	69.4 °C	90.0 °C	6.4E-02	2.0E-01	7.28E-10 s	8.02E-10 s	6.3E+3 h	7.2E+2 h	7.5E+2 h	8.6E+1 h
log2	27460	1.45E-08 s	1.64E-08 s	1.68E-08 s	75.3 °C	90.0 °C	8.9E-02	2.0E-01	1.80E-08 s	1.94E-08 s	8.0E+3 h	1.3E+3 h	9.5E+2 h	1.6E+2 h
max	3096	8.69E-09 s	9.64E-09 s	9.84E-09 s	79.8 °C	90.0 °C	1.1E-01	2.0E-01	1.05E-08 s	1.11E-08 s	1.6E+4 h	3.3E+3 h	1.9E+3 h	3.9E+2 h
mem_ctrl	26294	4.05E-09 s	4.26E-09 s	4.75E-09 s	84.8 °C	90.0 °C	1.5E-01	2.0E-01	4.48E-09 s	5.51E-09 s	8.6E+4 h	8.6E+2 h	8.6E+4 h	1.0E+2 h
multiplier	28525	1.06E-08 s	1.19E-08 s	1.22E-08 s	87.2 °C	90.0 °C	1.7E-01	2.0E-01	1.34E-08 s	1.40E-08 s	4.6E+3 h	1.5E+3 h	5.5E+2 h	1.8E+2 h
priority	485	1.76E-09 s	1.95E-09 s	2.00E-09 s	32.3 °C	43.3 °C	5.9E-03	1.3E-02	1.97E-09 s	2.05E-09 s	3.1E+5 h	4.7E+4 h	3.6E+4 h	5.6E+3 h
router	79	5.55E-10 s	6.31E-10 s	6.46E-10 s	32.6 °C	44.2 °C	6.0E-03	1.3E-02	6.37E-10 s	6.70E-10 s	9.6E+4 h	1.6E+4 h	1.1E+4 h	1.9E+3 h
sin	4147	6.93E-09 s	7.77E-09 s	7.96E-09 s	33.5 °C	46.7 °C	6.4E-03	1.6E-02	7.85E-09 s	8.27E-09 s	1.6E+5 h	2.3E+4 h	1.9E+4 h	2.7E+3 h
sqrt	12859	1.97E-07 s	2.16E-07 s	2.20E-07 s	33.1 °C	45.6 °C	6.2E-03	1.5E-02	2.18E-07 s	2.27E-07 s	5.0E+5 h	7.7E+4 h	5.9E+4 h	9.2E+3 h
square	17743	9.47E-09 s	1.05E-08 s	1.07E-08 s	33.7 °C	47.5 °C	6.5E-03	1.7E-02	1.06E-08 s	1.11E-08 s	3.2E+5 h	4.0E+4 h	3.8E+4 h	4.8E+3 h
voter	7528	2.17E-09 s	2.42E-09 s	2.49E-09 s	35.7 °C	53.7 °C	7.5E-03	2.5E-02	2.45E-09 s	2.63E-09 s	1.9E+5 h	1.4E+4 h	2.3E+4 h	1.6E+3 h



(a) Delay increase due to BTI-aging with consideration of temperature

(b) Time until 15% guard-band is missed

(c) Time until 10% guard-band is missed

Fig. 2: Impact of temperature on degradation. Arrows indicate values outside axis scale

That means, in the given technology one can expect a delay degradation of over 15% in a 10 years life period for random applications. These results confirm similar values reported in the literature, e. g. [9]. The application of input vectors coming from the *HotAging* algorithm increased the delay by up to 18% (arbiter) and in average by 15%. That means, in worst cases or if maliciously intended, the delay increase can be almost 20%. It should be emphasized that these values are only achieved if it assumed the circuits runs only at low temperature. However, if normal operation temperatures are considered then degradation can be drastically increased as reveled by Fig. 2a. It shows that for randomized input vectors the degradation increases for random values in average by 15% and up to 26% (multiplier), while for input vectors generated by the *HotAging* algorithm the delay increase by up to 37% (int2float) and in average by 22%.

In order to explore the applicability of the common anti-aging approach guard-banding, we defined a maximum delay increase of 15% and 10%, respectively. If the maximum circuit delay goes beyond these values, the safety of the circuit cannot be guaranteed anymore and it must be considered as malfunctioned. Figs. 2c and 2b depict for random input vectors and vectors generated by the *HotAging* algorithm the time until the maximum circuit delay exceeds the guard-band. In case of random input vectors the determined average time is above 10 years for 15% guard-band, but only 3.3 years for a 10% guard-band. Some circuits, though, reach the 15% guard-band already after less than a year and the 10% guard-band in nearly a month (e. g. int2float and multiplier). Results get even more severe, if vectors generated by the *HotAging* algorithm are considered. Here, the 15% guard is missed in average in 5 years, with minimum values below two months (e. g. int2float and mem_ctrl). In case of the 10% guard, the average time until failure is 7.1 months, while several circuits would fail within 2 weeks. One can clearly state, worst case aging, which can be happen coincidently or maliciously intended, can seriously impact the reliability of an integrated circuit.

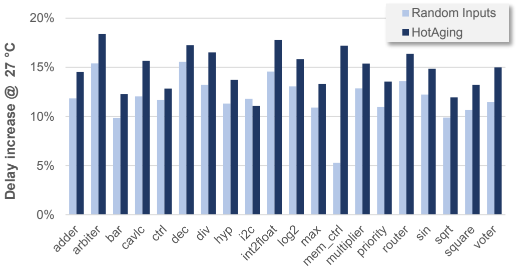


Fig. 3: Delay increase due to BTI-aging without consideration of temperature

IV. Conclusions

In this work, we explored the relation between hardware degradation and temperature. Therefore, we implemented an analysis environment that permits the extraction of random and worst case input scenarios and the determination of how aging and temperature impact the circuit delay. Results indicate that BTI-aging without consideration of temperature can increase circuit delay in average by 12% (random case) and 15% (worst case) in a 10 years life period, confirming already reported results. However, if temperature is considered degradation can increase to 26% (random case) or even nearly 40% (worst case). If guard-banding is applied, i. e. an acceptable delay degradation is defined, circuits can enter malfunction states within months (random case) or even within weeks (worst case). These results emphasize the need for appropriate countermeasures that consider the actual aging effects together with the accelerating influence of high temperatures.

References

[1] K. Jeong, A. B. Kahng, and K. Samadi, “Impact of guardband reduction on design outcomes: A quantitative approach,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 22, no. 4, pp. 552–565, Nov 2009.

[2] C. Niemann, T. Wegner, D. Timmermann, and F. Sill Torres, “Low overhead in situ aging monitoring and proactive aging management,” in *IEEE International Symposium on Circuits and Systems*, 2016, pp. 2799–2802.

[3] S. Das, C. Tokunaga, S. Pant, W. Ma, S. Kalaiselvan, K. Lai, D. Bull, and D. Blaauw, “Razorii: In situ error detection and correction for pvt and ser tolerance,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, p. 32–48, January 2009.

[4] F. Sellers, *Error detecting logic for digital computers*. McGraw-Hill, 1968.

[5] R. Tu *et al.*, “Berkeley reliability tools-bert,” *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 12, no. 10, p. 1524–1534, Feb 2093.

[6] F. Sill Torres, P. F. R. Leite, and R. Drechsler, “Unintrusive aging analysis based on offline learning,” in *2017 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Oct 2017, pp. 1–4.

[7] S. Kiamehr, F. Firouzi, M. Ebrahimi, and M. B. Tahoori, “Aging-aware standard cell library design,” in *Design, Automation and Test in Europe*, March 2014, pp. 1–4.

[8] N. Karimi, A. K. Kanuparthi, X. Wang, O. Sinanoglu, and R. Karri, “Magic: Malicious aging in circuits/cores,” *Trans. Archit. Code Optim.*, vol. 12, no. 1, pp. 5:1–5:25, apr 2015.

[9] H. M. Abbas, B. Halak, and M. Zwolinski, “Bti mitigation by anti-ageing software patterns,” *Microelectronics Reliability*, vol. 79, pp. 79 – 90, 2017.

[10] “Reliability report 1h2017,” Intel, Tech. Rep., 2017.

[11] W. Huang, K. Sankaranarayanan, K. Skadron, R. J. Ribando, and M. R. Stan, “Accurate, pre-rtl temperature-aware design using a parameterized, geometric thermal model,” *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1277–1288, Sept 2008.

[12] W. Zhao and Y. Cao, “New generation of predictive technology model for sub-45 nm early design exploration,” *IEEE Transactions on Electron Devices*, vol. 53, no. 11, pp. 2816–2823, Nov 2006.

[13] K. Joshi, S. Mukhopadhyay, N. Goel, and S. Mahapatra, “A consistent physical framework for n and p bti in hkmg mosfets,” in *International Reliability Physics Symposium (IRPS)*, April 2012, pp. 5A.3.1–5A.3.10.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[15] R. Salomon and F. Sill, “High-speed, low-leakage integrated circuits: An evolutionary algorithm perspective,” *Journal of Systems Architecture*, vol. 53, no. 5, pp. 321 – 327, 2007.

[16] T. Back, U. Hammel, and H. . Schwefel, “Evolutionary computation: comments on the history and current state,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, April 1997.

[17] I. Rechenberg, *Evolutionsstrategie*. Frommann-Holzboog, Stuttgart, 1994.

[18] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. USA: Addison-Wesley Publishing Company, 2010.

[19] E. J. Wyrwas and J. B. Bernstein, “Quantitatively analyzing the performance of integrated circuits and their reliability,” *IEEE Instrumentation Measurement Magazine*, vol. 14, no. 1, pp. 24–31, Feb 2011.

[20] K. Gray and J. Paschkewitz, *Next Generation HALT and HASS: Robust Design of Electronics and Systems*, ser. Quality and Reliability Engineering Series. Wiley, 2016.

[21] F. Sill, F. Grassert, and D. Timmermann, “Reducing leakage with mixed-v/sub th/ (mvt),” in *18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design*, Jan 2005, pp. 874–877.

[22] L. Amarú, P.-E. Gaillardon, and G. De Micheli, “The EPFL combinational benchmark suite,” in *Int’l Workshop on Logic Synth.*, 2015.