Reliability-Aware Design to Suppress Aging

Hussam Amrouch*, Behnam Khaleghi†, Andreas Gerstlauer‡ and Jörg Henkel*
*Karlsruhe Institute of Technology, Karlsruhe, Germany, †Sharif University of Technology, Tehran, Iran, ‡University of Texas, Austin, USA
{amrouch; henkel}@kit.edu, behnam_khaleghi@ce.sharif.edu, gerstl@ece.utexas.edu

ABSTRACT

Due to aging, circuit reliability has become extraordinary challenging. Reliability-aware circuit design flows do virtually not exist and even research is in its infancy. In this paper, we propose to bring aging awareness to EDA tool flows based on so-called degradation-aware cell libraries. These libraries include detailed delay information of gates/cells under the impact that aging has on both threshold voltage (V_{th}) and carrier mobility (μ) of transistors. This is unlike state of the art which considers V_{th} only. We show how ignoring μ degradation leads to underestimating guardbands by 19% on average. Our investigation revealed that the impact of aging is strongly dependent on the operating conditions of gates (i.e. input signal slew and output load capacitance), and not solely on the duty cycle of transistors. Neglecting this fact results in employing insufficient guardbands and thus not sustaining reliability during lifetime.

We demonstrate that degradation-aware libraries and tool flows are indispensable for not only accurately estimating guardbands, but also efficiently *containing them.* By considering aging degradations during logic synthesis, significantly more resilient circuits can be obtained. We further quantify the impact of aging on the degradation of image processing circuits. This goes far beyond investigating aging with respect to path delays solely. We show that in a standard design without any guardbanding, aging leads to unacceptable image quality after just one year. By contrast, if the synthesis tool is provided with the degradation-aware cell library, high image quality is sustained for 10 years (even under worst-case aging and without a guardband). Hence, using our approach, aging can be effectively suppressed.

1. INTRODUCTION

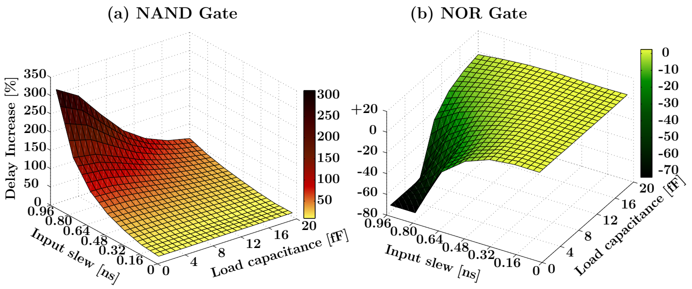
Technology scaling is reaching limits where displacing a few atoms within transistors due to aging phenomena may endanger the functionality of the entire design. Negative and Positive Bias Temperature Instability (NBTI and PBTI) are the most prominent of these phenomena, with the potential to remarkably degrade the electrical characteristics of pMOS and nMOS transistors, respectively. They occur due to electrical field stresses in transistors resulting in interface traps (when *Si-H* bonds break at the *Si-SiO*$_2$ interface) and oxide traps (when charges are captured in the oxide vacancies within the dielectric). Over time, generated defects (i.e. interface/oxide traps) accumulate inside the transistor, manifesting themselves as degradations in its electrical characteristics (V_{th}, μ, etc.). Hence, aged transistors become slower, which increases the likelihood of timing violations. BTI has also instantaneous effects in which degradations are observed within a very short time domain (e.g., $1\mu s$) [2]. However, this work considers only the long-term effects of aging.

Timing Guardband: To overcome aging, manufacturers typically employ a *guardband* (T_G) on top of the critical path delay (T) of a design [3] to guarantee that it will always be clocked at a sustainable frequency for the projected lifetime. $f = \frac{1}{T(t=lifetime)}$; $T(t = lifetime) = T(t = 0) + T_G$. Recent technology nodes operating under decreased supply voltages steadily narrow the available design space for these guardbands. Thus, minimizing guardbands becomes an inevitable design task that additionally needs to be considered. Traditionally, guardbanding is treated as a post-synthesis optimization. However, to avoid large guardbands (which may not be tolerated any longer), aging degradations should be considered during the logic synthesis itself in order to obtain circuits that are inherently more resilient against aging.

Commercial EDA flows have evolved over three decades of research. They provide powerful capabilities for analyzing the timing behavior of circuits and thus precisely determining their critical path delay based on information provided within the targeted cell library. In addition, logic synthesis can (also based on delay information within the targeted cell library) efficiently optimize the circuit's netlist to maximize the performance. Therefore, bringing reliability awareness to tool flows is essential not only to accurately estimate guardbands (i.e. without under/over-estimation) but also to optimize the circuits against aging and thus *containing* guardbands (i.e. having small, yet sufficient guardbands). In this paper, we propose to provide the synthesis tool with the degradation-aware cell library to allow it addressing aging concerns – even if it was not designed for that purpose.

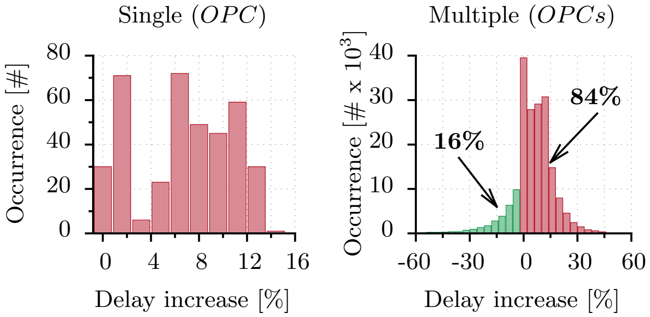
Our main contributions are as follows:
(1) We explore the role of operating conditions of a gate/cell in determining the overall impact of aging in the scope of both timing analysis and logic synthesis – this holds even more for complex designs like processors.
(2) By providing logic synthesis tools with the delay information of gates/cells under aging, we leverage mature optimization algorithms to effectively suppress aging effects.
(3) To achieve our goals, we created 121 degradation-aware cell libraries under varied aging scenarios. Libraries are *publicly available at* [1] and ready to be used with existing tool flows (e.g., Synopsys) without requiring any modifications.

DAC'16, June 05-09, 2016, Austin, TX, USA
© 2016 ACM. ISBN 978-1-4503-4236-0/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2897937.2898082


(a) NAND Gate

(b) NOR Gate

Figure 1: Impact of aging on the delay of NAND and NOR gates demonstrating how it is *driven* by its operating conditions.


Single (OPC)

Multiple ($OPCs$)

Figure 2: Considering a single operating condition (OPC) solely, instead of multiple $OPCs$, leads to erroneous aging estimations.

2. MOTIVATIONAL ANALYSIS: LINKING THE PHYSICAL AND SYSTEM LEVELS

In the following, we briefly explain how aging-induced defects at the physical level may propagate all the way up to the system level where they finally manifest themselves as timing errors. The complexity of having to jointly consider multiple levels motivates the necessity of addressing aging within EDA tools to effectively optimize reliability. We obtained our results in Figs. (1, 2 and 3) from our proposed methods in Section 4 which presents the core of this work.

1) Physical/Device-Level: During the operation of a transistor, aging-induced interface traps interact with the charge carriers within the channel degrading its mobility (μ). In addition, interface and oxide traps jointly cause a charge buildup that shifts the threshold voltage (V_{th}). A transistor's *duty cycle* (λ), which indicates the proportion of time that the transistor is in stress thereby determines the overall number of defects and thus the (V_{th}, μ) degradations over time. λ takes a value between 1 (indicating *worst-case* aging) and 0 (indicating that aging did not occur). It is noteworthy that aging optimization techniques often aim to drive the λ of transistors towards the *balance-case* of 0.5.

2) Gate-Level: Due to interdependencies between the electrical characteristics of a MOSFET, degradations in V_{th} and μ lead to degrading other characteristics like the drain current (I_d) which is the cause of aged transistors becoming slower, as the first order approximation in Eq. 1 illustrates. The detailed modeling that we employ can be found in [5].

$$Delay \propto \frac{1}{I_d}\ ; \qquad I_d \approx \frac{\mu}{2}\cdot(V_{dd}-V_{th}-\Delta V_{th})^2 \qquad (1)$$

Operation Conditions ($OPCs$): Each gate within a circuit's netlist is subject to different $OPCs$ based on the slew of its input signals and the load capacitance of its output. The input slew is determined by the output slope of all previous gates that are connected to that particular gate. By contrast, the number and types of gates that are connected to the output of a gate determine the load capacitance of that gate. Our investigation revealed that $OPCs$ play a major role in determining the impact that aging has on a gates' delay increase. Importantly, our results also revealed that under some $OPCs$, the delay of some gates improves instead of being degraded. Figs. 1(a, b) show the impact of aging on two different gates (NAND and NOR, respectively). As is shown in Fig. 1(a), a larger input signal slew magnifies the impact of aging on increasing the NAND's delay as it leads to activating both the pull-up (pMOS) and pull-down (nMOS) networks of the NAND simultaneously. Since the degradations of NBTI in pMOS transistors are higher than the degradations of PBTI in nMOS transistors [6], the pull-down network becomes relatively stronger and thus opposes the pull-up network – especially for larger slews in which the "ON" periods of pMOS and nMOS transistors overlap further. On the other hand, increasing the output load capacitance diminishes the impact of aging. Since it makes the NAND slower, it allows the gate to tolerate the degradations in its pMOS and nMOS transistors. The NOR gate exhibits a different behavior (Fig. 1(b)). Here, NBTI degradations make the opposing current from the pull-up network become smaller and hence the fall delay improves[1] – especially for larger input slews. Note that even though aging may improve the delay of some gates under specific $OPCs$, it will still make the overall circuit slower. The high increase in the delay of other gates within each path will usually compensate the small decrease in some individual gates. Fig. 2 summarizes the impact that worst-case aging (i.e. $\lambda = 1$) has on a standard cell library. When only a single OPC (e.g., the slowest signal slew along with the smallest output capacitance) is considered, aging leads to degrading all the gates' delays (by up to 15%). When multiple $OPCs$ are considered, the impact of aging on gate delays exhibits a significantly wider range (between -60% and 400%), and the likelihood that aging improves a gate's delay reaches 16%.

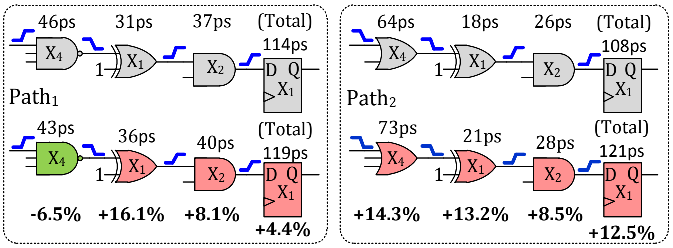


Figure 3: Motivational example showing how path$_1$, which was critical before aging, became uncritical after aging.

Therefore, it becomes evident that $OPCs$ play a major role. Targeting only a single OPC results in erroneously estimating the overall impact of aging on the paths of circuits.

3) Circuit-Level: Since different gates under different $OPCs$ exhibit varied timing behaviors due to aging, it can be expected that the critical path (CP) of a circuit may become uncritical after aging and another path, which formerly was uncritical, may afterwards become critical. Fig. 3 presents a realistic example (i.e. all presented delays are measured by HSPICE) of two paths, where aging switches their roles with respect to criticality. In spite of both paths identically suffering from aging (i.e. worst-case aging is applied to all transistors), the impact of aging on the overall delay of each path is different. This is due to paths starting with different gates and thus different signal slopes propagating through each path. Consequently, every gate will have its own OPC and will be differently influenced by aging degradations. *Hence, optimizing aging is not only a matter of bal-*

[1] It is only valid in the case of input rise. Hence, one cannot initially make the pMOS weaker to improve the gate's delay.

ancing the duty cycle of transistors as state of the art often assumes. It is rather a question of selecting the most suitable gate with respect to aging, based on the existing $OPCs$.

4) System-Level: Even though aging degradations originate at the physical level, their observable effects finally appear at the system level, manifesting themselves as errors due to timing violations. Importantly, since aging degrades the paths of circuits differently, *interpreting aging* (i.e. quantifying the overall impact of aging on the design's reliability) cannot be determined through studying the critical path solely. Instead, all potentially violated paths need to be jointly considered as each one contributes its share to overall reliability degradation.

3. RELATED WORK

In the context of this paper, previous work can be divided into the following two categories:

Aging Estimation: In [7], mathematical models are employed to correlate the NBTI-induced ΔV_{th} in pMOS transistors to the corresponding gate delay increase. In addition to neglecting the PBTI effects in nMOS transistors, the mathematical model itself is not able to consider the *slope* of rise and fall signals of a cell. Therefore, this method fails to analyze multi-stage cells (e.g., buffers, flip-flops, etc.) and multi-gate paths. In fact, multi-stage cells may form $> 50\%$ of a standard cell library [8]. Hence, neglecting them leads to a severe impact on aging estimation. [9] suffers may from a similar drawback. It can also not analyze multi-stage cells as it fails to consider the slope of internal signals inside such cells. [10] proposed to measure the impact of NBTI-induced ΔV_{th} on a gate's delay. However, it neglects PBTI. This leads to inaccurate analysis as PBTI degradations may compensate/magnify NBTI degradations. [11] accurately estimates the effects of NBTI and PBTI using HSPICE. However, the impact of aging on degrading μ was not considered.

When complex designs like processors are targeted, state-of-the-art approaches often analyze the impact of aging on the CP only (e.g., [13]). However, aging may switch a path from critical to uncritical and vice versa (see Section 2 and Fig. 3). Thus, looking solely at the CP is not sufficient. Other works (e.g., [12]) proposed to consider the top $x\%$ of CPs. This might not be feasible in realistic designs. The number of paths within the top 5% may reach $> 10^7$ [14]. In practice, determining an x such that it is guaranteed that the path that may become critical after aging is included is not trivial. Importantly, [12, 13] neglect PBTI and they only consider a single OPC, which leads to erroneously estimating aging effects, as earlier demonstrated in Figs. (2, 1).

It is noteworthy that aging degrades both V_{th} and μ [15]. However, state of the art considers only V_{th} in their analysis. Our evaluation results in Section 5 demonstrate that neglecting the μ degradation leads to underestimating the required guardbands by 19%. Thereby, both V_{th} and μ need to be *jointly* considered to accurately estimate guardbands.

Aging Optimization: [14] introduced a technique to optimize the circuit against aging by capturing potential paths that may become critical after aging, and then iteratively applying tighter timing constraints on them in order to force the synthesis tool to optimize these paths. The key drawback is that in each iteration and with each new constraint, the structure of paths may change and thus different gates will be used. However, the new gates might be more susceptible to aging and thus the new paths might age faster than the former ones. Finally, to mitigate aging effects, there are many techniques (e.g., [4]) that aim to balance the *duty cycle*. In general, such techniques are orthogonal to our work. Importantly, the impact of aging is not only a function of the *duty cycle* but, additionally, of the $OPCs$ as presented in Figs (1, 2). Hence, the effectiveness of employing aging-balancing techniques *standalone* might be limited.

Distinction from existing state of the art:
(1) We consider the impact of aging on both V_{th} and μ as well as the impact of aging under multiple $OPCs$.
(2) We integrate circuits optimization against aging effects within the standard design flow through plugging into the existing tool flows the degradation-aware cell library.
(3) We investigate the overall impact of aging on reliability through interpreting how aging manifests itself at the system level in the scope of image processing. *This goes far beyond investigating aging with respect to path delays solely.*

4. RELIABILITY-AWARE DESIGN

Concisely, our idea is to create degradation-aware cell libraries under different aging stress scenarios. Plugging the required library into a timing analysis tool will enable accurate analysis of the timing behavior of the entire circuit in the scope of static and/or dynamic aging stress. Hence, the required guardband can be obtained. Additionally, synthesizing the circuit based on the degradation-aware cell library enables optimization algorithms to consider aging effects (in every gate/cell) and thus select the most suitable gate/cell for each OPC. Fig. 4 gives a general overview of our work.

4.1 Degradation-Aware Cell Libraries

To accurately model aging effects, we start at the lowest physical level where BTI does occur. We employ recently proposed physics-based models [6] to estimate the number of generated defects within nMOS and pMOS transistors. In addition to their accuracy, unlike empirical models used by others (e.g., [12, 13]), where only V_{th} is considered, physics-based aging models allow for estimating both V_{th} and μ degradations. This can be done similar to e.g., [15]:

$$\Delta V_{th} = \frac{q}{C_{ox}} \cdot (\Delta N_{IT} + \Delta N_{OT}) \tag{2}$$

$$\Delta\mu = \frac{\mu_0}{1 + \alpha \cdot \Delta N_{IT}} \tag{3}$$

There, N_{IT} and ΔN_{OT} are the number of generated interface and oxide traps, respectively, obtained from [6]. Note that the employed aging model does not consider the intrinsic variability of BTI. However, one could get the distribution of ΔV_{th} from e.g., [16] and then select the worst-case degradation (e.g., 6σ) as the upper bound.

Afterwards, we create a *degraded* transistor model for a given technology library model (i.e. based on an unmodified model that contains initial transistor characteristics) by changing V_{th} and μ in both pMOS and nMOS transistors according to the estimated aging degradations. Since aging degradations depend on the *duty cycle* (λ), we repeat estimations for N different λ cases of both pMOS and nMOS transistors resulting in $N \times N$ *degraded* transistor models. Then, we create the corresponding *degradation-aware* cell library for each obtained model (see Fig. 4(a)) as follows.

We first measure the delay and slope of each gate/cell within the library under different $OPCs$ using HSPICE and the *degraded* transistor model[2] along with the SPICE definition of that gate/cell. The latter includes the required

[2] For simplicity we assume that all pMOS in a gate have the same λ_{pmos} and similarly all nMOS have the same λ_{nmos}. Due to the vast variety of activities over time, this can be reasonable. However, our method is not limited to this assumption and it can analogously be applied for others.

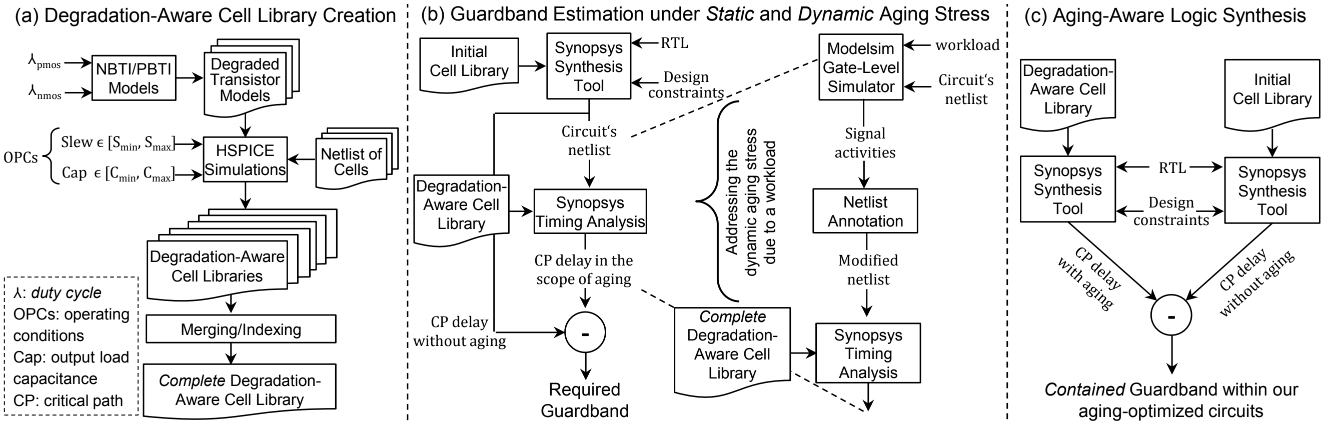

(a) Degradation-Aware Cell Library Creation
λ_{pmos}
λ_{nmos}
NBTI/PBTI Models
Degraded Transistor Models
OPCs
Slew ϵ [S_{min}, S_{max}]
Cap ϵ [C_{min}, C_{max}]
HSPICE Simulations
Netlist of Cells
Degradation-Aware Cell Libraries
Merging/Indexing
Complete Degradation-Aware Cell Library
λ: duty cycle
OPCs: operating conditions
Cap: output load capacitance
CP: critical path
(b) Guardband Estimation under Static and Dynamic Aging Stress
Initial Cell Library
Synopsys Synthesis Tool
RTL
Design constraints
Circuit's netlist
Degradation-Aware Cell Library
Synopsys Timing Analysis
CP delay in the scope of aging
CP delay without aging
-
Required Guardband
Addressing the dynamic aging stress due to a workload
Modelsim Gate-Level Simulator
workload
Circuit's netlist
Signal activities
Netlist Annotation
Modified netlist
Complete Degradation-Aware Cell Library
Synopsys Timing Analysis
(c) Aging-Aware Logic Synthesis
Degradation-Aware Cell Library
Initial Cell Library
Synopsys Synthesis Tool
RTL
Design constraints
Synopsys Synthesis Tool
CP delay with aging
CP delay without aging
-
Contained Guardband within our aging-optimized circuits

Figure 4: A general overview of our reliability-aware circuit design flow that integrates automatic aging optimizations.

parasitics information (i.e. inner capacitances and resistors) based on the gate/cell layout, which allows for more accurate analysis. The *OPCs* are defined as a range of input signal slews $[S_{min}, S_{max}]$ along with a range of output load capacitances $[C_{min}, C_{max}]$. The required boundaries are determined as follows: since S_{min} is the minimum slew, it needs to be set based on the fastest gate in the library under no fanout. By contrast, S_{max} is the maximum slew and therefore set based on the slowest gate connected to the maximum fanout. C_{min} and C_{max} are determined based on the smallest (i.e. no fanout) and largest (i.e. maximum allowed fanout) output capacitance, respectively, of the smallest gate. Finally, we merge all the resulting degradation-aware libraries into one *complete* library. To do so, we add corresponding indexes that distinguish between identical cells. For example, two AND2 gates with two different aging stress cases ($\lambda_{PMOS} = 0.4$, $\lambda_{NMOS} = 0.6$) and ($\lambda_{PMOS} = 0.9$, $\lambda_{NMOS} = 0.5$) will be renamed to AND2_0.4_0.6 and AND2_0.90_0.5, respectively. In practice, the *complete* degradation-aware cell library contains the delay of each cell under $(N \times N)$ different aging stresses.

4.2 Estimating Guardbands

After synthesizing the circuit with the degradation-unaware (i.e. initial) cell library to generate the netlist, we can estimate the impact of aging either under *static* (i.e. all transistors having the same λ) or *dynamic* (i.e. each transistor is influenced by aging according to the running workload) stress scenarios to estimate the required guardband.

Static Aging Stress: Here, the timing analysis tool reads the netlist along with the degradation-aware cell library that corresponds to the wanted aging stress case to report the path delays. Through comparing the initial delay of the circuit (i.e. the delay of the initial CP before aging) with the new obtained delay in the scope of aging (i.e. the delay of the possibly new CP after aging), the required guardband that is sufficient to protect the circuit against this particular aging stress can be determined (see Fig. 4(b)).

Dynamic Aging Stress: Here, the signal probability profile caused by the workload needs to be first estimated using a gate-level simulator. This provides the activity and thus *duty cycle* (λ) of each individual transistor within the netlist. Afterwards, the average *duty cycles* for nMOS and pMOS transistors in each gate/cell within the netlist are calculated. The netlist is then modified by annotating each gate/cell with the indexes of $Avg(\lambda_{nmos})$ and $Avg(\lambda_{pmos})$. For instance, an AND2 gate instance with $Avg(\lambda_{nmos}) = 0.4$ and $Avg(\lambda_{nmos}) = 0.6$ will be renamed to AND2_0.4_0.6. This is necessary for compatibility with the format of the complete degradation-aware library that we created in Section 4.1. Finally, the timing analysis tool reads the modified netlist along with the complete degradation-aware library, which contains the delay under different λ cases for each gate/cell, to report the path delays. It is noteworthy that such an aging analysis is only valid for that particular workload. Other workloads may cause different activities and thus a new analysis will indeed be required.

To suppress aging under any workload (i.e. avoiding the workload dependency), static *worst-case* aging can be considered (i.e. λ_{PMOS}=1.0, λ_{NMOS}=1.0) during the analysis to estimate the required guardband.

4.3 Containing Guardbands

As demonstrated in Section 2, the impact of aging on the delay of a gate is subject to its operating conditions. Therefore, knowing how each gate/cell (with different *OPCs*) behaves allows for the synthesis tool to always select the most suitable gate/cell when minimizing the delay of paths in an aged circuit. As observed in Fig. 1(a), the impact of aging on the NAND gate, for instance, would be alleviated when the output capacitance is high. Similarly, when the input slew is smaller, the aging impact becomes lower. Hence, the synthesis algorithms would select this gate type when a high output load (i.e. high fanout) exists. Likewise, the tool could use input buffers to sharpen the slew of input signals. Practically, if the synthesis tool is provided with the degradation-aware library, it becomes *aware* of the impact of aging on the gates/cells' delays. Thus, it can automatically select the most suitable gate type for each OPC towards optimizing the aged circuit's netlist. In this work, we employ our degradation-aware cell library assuming *worst-case* aging to let synthesis optimize the netlist independent of the workload. Note that the obtained CP delay here is already in the presence of aging and hence no additional guardband needs to be applied. However, the *included* guardband can be computed as the timing difference between analyzing our design with the initial (i.e. degradation-unaware) and degradation-aware libraries. An equivalent *contained* guardband is computed as the performance penalty when synthesizing for aging by comparing the obtained CP delay against the CP delay of a traditionally-optimized version of the design synthesized with the initial cell library (see Fig. 4(c)).

4.4 Implementation and Design Details

At the device-level: The high-performance 45 nm Predictive Technology Model (PTM) [17] is used for both nMOS and pMOS transistors. These models are designed for a high-k technology and thus compatible with the employed

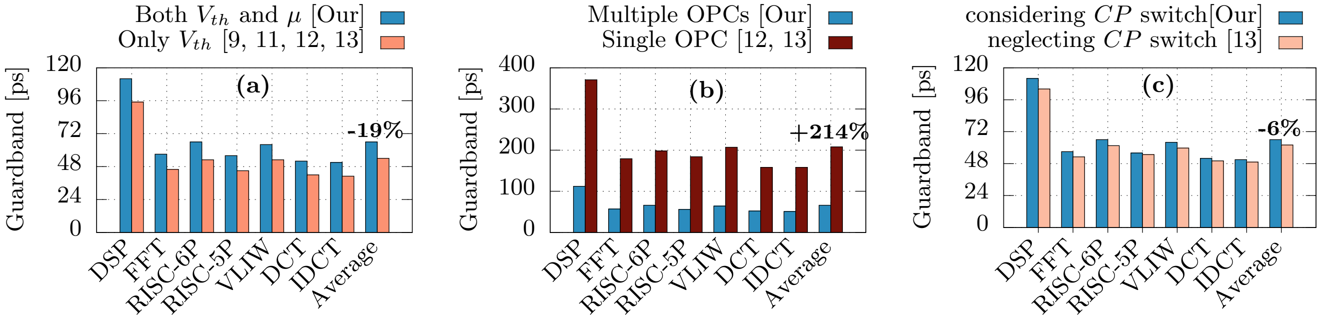


Figure 5: Evaluation of guardband estimation. The impact of neglecting μ and *OPCs* are shown in (a) and (b), respectively. (c) shows the necessity of considering the actual path that will become critical after aging instead of the initial path that formerly was critical.

aging models [6]. The λ ranges from the case of *no-aging* to the *worst-case* (i.e. $\lambda \in [0, 1]$) with a step of 0.1. Thus, 121 degradation-aware cell libraries are created, in addition to the final *complete* library after merging all of them.

At the gate-level: The open-source Nangate 45 nm standard cell library [8] is employed to obtain realistic netlists of different 68 combinational and sequential gates/cells. Parasitics information is included based on layout at the 45 nm node. Other works (e.g., [13, 12]) neglect parasitics leading to inaccurate analysis. We considered 49 *OPCs* for each gate (7 input signal slews and 7 output load capacitances). This range is consistent with what is used in the Nangate library [8], and it is also typical in industrial libraries such as Synopsys' 32/28 nm. The S_{min} and S_{max} are set to 5 ps and 947 ps, respectively. In addition, the C_{min} and C_{max} are set to 0.5 fF and 20 fF, respectively (details in Section 4.1). A V_{dd} of 1.2 V is considered, and HSPICE along with BSIM 4.0 is used to measure gates delays. Employing BSIM modeling allows us to consider the interdependencies of the electrical characteristics of MOSFET and hence how V_{th} and μ degradations influence other parameters.

At the circuit-level: The Design Compiler from Synopsys is used to synthesize the circuits and to perform the required timing analysis. During synthesis, the *compile ultra* option is used to optimize the designs along with the *highest effort* along with an objective of performance maximization. Modelsim is used to estimate the transistors' *duty cycles* (i.e. the *dynamic* aging stress) by extracting activities of running workloads on top of the design's netlsit.

5. EVALUATION AND COMPARISON

To demonstrate the benefits of our work in optimizing the reliability of complex circuits, we employed 5 different kinds of processor designs: VLIW, RISC (with 5 and 6-pipeline stages), FFT and DSP. In addition, we also included DCT and IDCT circuits used in image processing. The average area of our circuits is 4x larger than the IWLS benchmarks [18] which are often used in other works. We purposefully chose them to evaluate the effectiveness of our work in industrial-strength designs.

Guardband Estimation: As explained in Section 2, aging-induced defects at the physical level degrade both V_{th} and μ of transistors at the device level. Such degradations then alter the gates' delays based on the encountered *OPCs*. Finally, the overall delay of every path, at the circuit level, changes based on the new delays of each individual gate. As illustrated in Section 2, the path that was initially (i.e. before aging) critical may not remain critical after aging. State-of-the-art works consider only V_{th} [9, 11, 12, 13], a single *OPC* [12, 13] or may not take the impact of aging on switching the circuit's CP into account [13]. Fig. 5 quantifies the impact of doing that on the resulting guardbands, where estimations were performed under worst-case aging for a lifetime of 10 years. Each of (a, b and c) in Fig. 5 evaluates the impact of neglecting a single aspect (μ, different *OPCs* or CP switching, respectively) alone to allow fair comparisons and to quantify the role of each aspect individually.

As shown in Fig. 5(a), neglecting the μ degradation leads to under-estimating the required guardband by 19% on average. Fig. 5(b) demonstrates the severe impact of neglecting different *OPCs*, where the guardband may be over-estimated by 214% on average. Additionally, Fig. 5(c) shows that only considering the initial CP when estimating the guardband provides a wrong guardband in all circuits.

Aging Optimization: As illustrated in Section 4.3, providing the synthesis tool with our degradation-aware cell library allows for the optimization algorithms to consider the impact of aging on gates/cells while synthesizing the circuit. As a result, the netlist will proactively be more resilient to aging and the guardband will be inherently contained. Fig. 6(a) compares the *required guardband* when synthesizing with the initial (i.e. degradation-unaware) library and the *contained guardband* when synthesizing with our degradation-aware cell library. This provides a direct comparison between traditionally-optimized designs with guardbands and our aging-optimized designs with *contained* guardbands relative to the same baseline. Our aging-optimized designs show on average 50% and up to 75% smaller guardbands leading to 4% and up to 6% higher frequency. This comes with merely 0.2% area overhead as shown in Fig. 6(b).

Suppressing aging at system-level: Finally, we study the impact of aging within DCT and IDCT circuits used to first encode and then decode images. Because image processing circuits can inherently tolerate errors, we select them to evaluate the impact of our optimizations on suppressing aging *when no guardband is employed*. This links the physical and system level by quantifying how defects generated within transistors affect the overall reliability represented by image quality. Fig. 6(c) reports the Peak Signal to Noise Ratios (PSNRs) of a fixed-point DCT-IDCT image processing chain under different scenarios (a PSNR of 30dB is typically considered an acceptable image quality). Note that adding a guardband would lead to tolerating the resulting degradations and thus the PSNR would not drop. All results obtained from gate-level simulations which employ the "sdf" files of DCT/IDCT generated from the synthesis tool under the targeted aging scenario (e.g., worst/balance case).

To fairly quantify the aging impact on image quality, we used the same frequency to evaluate all studied scenarios (i.e. *all gate-level simulations were run at the same frequency*). This frequency was selected based on the maximum achieved performance in the *absence of aging* (i.e. when synthesizing

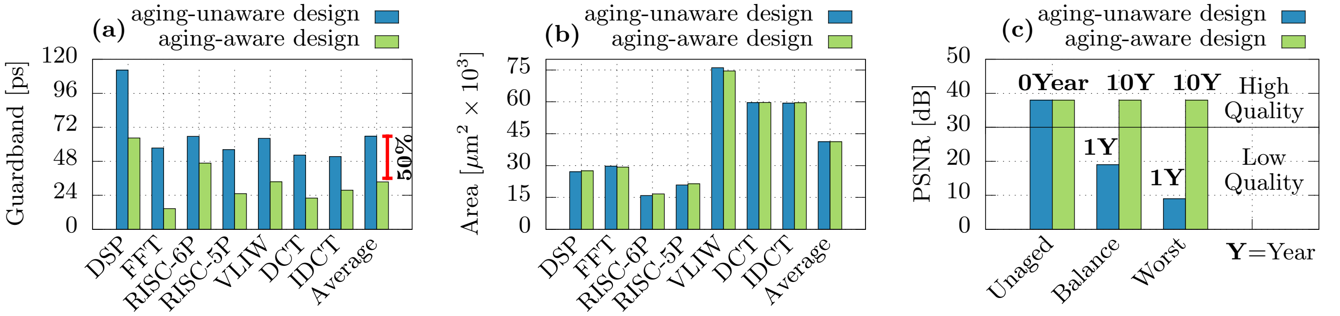


Figure 6: Effectiveness of synthesizing circuits with the degradation-aware cell library (*as our focus*): (a) Evaluates the reduction in the guardbands, (b) area overhead, (c) PSNR of DCT-IDCT showing how it remains the same as in the absence of aging even after 10 years.

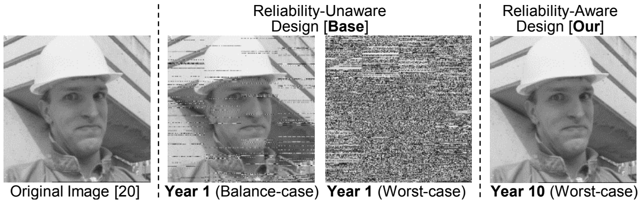

Reliability-Unaware Design **[Base]**
Reliability-Aware Design **[Our]**
Original Image [20]
Year 1 (Balance-case)
Year 1 (Worst-case)
Year 10 (Worst-case)

Figure 7: Outputs of the "DCT-IDCT" showing how our aging-aware design suppresses degradations for an extended lifetime.

the circuits with the initial cell library). Therefore, both aging-aware and aging-unaware designs will operate with no (*required* or *contained*) guardband. Note that an aging-aware design (without its *contained* guardband) may generally have timing errors even at year 0. However, several image examples (11) that we simulated did not result in any PSNR drop at year 0 compared to the aging-unaware design.

As seen in Fig. 7, **one year** of aging (under the worst-case) entirely destroys the image of an aging-unaware design (PSNR reaches 9dB). *This is due to not employing any guardband and thus degradation cannot be tolerated at all.* In fact, the synthesis tool aims during optimization at minimizing the delay of all paths along with balancing them. Thus, degradations here lead to a large number of timing errors, which severely degrade the image quality after a short lifetime. Even in the case of *balanced* aging – which is representative of state-of-the-art optimization techniques – degradations after one year are still able to noticeably reduce image quality (see Fig. 7) with the PSNR reaching 19dB. Thus, such techniques standalone are not sufficient.

By contrast, using the degradation-aware cell library (under worst-case) during synthesis results in circuits that effectively *suppress* aging effects. Even after **10 years** of worst-case aging, the PSNR remains the same as in the absence of aging. Hence, the lifetime (i.e. the time until the image quality drops to an unacceptable level of 30dB) increases by $>$ 10x because our aging-optimized circuit, after 10 years, retains a large margin. *Therefore, our aging-aware design operates for an extended lifetime even with no guardband.*

6. CONCLUSION

Bringing aging awareness to existing EDA tool flows is essential in obtaining reliable designs. We demonstrated the necessity of creating degradation-aware cell libraries and their impact on aging-aware circuit design. We showed how traditional design techniques lead to significant over-design of guardbands. We further demonstrated that providing the synthesis tool with the degradation-aware cell library results in resilient circuits in which aging effects are effectively suppressed. All in all, degradation-aware cell libraries enable reliability-aware design of circuits. They are publicly available to be directly deployed within existing tool flows [1].

Acknowledgments

This work is supported in parts by the German Research Foundation (DFG) as part of the priority program "Dependable Embedded Systems" [19] (SPP 1500 - spp1500.itec.kit.edu). Authors thank Amir Erfan Eshratifar and Ku He for their valuable support regarding the image processing evaluation.

7. REFERENCES

[1] "Degradation-Aware Cell Libraries, V1.0," **http://ces.itec.kit.edu/dependable-hardware.php**

[2] V. Santen, H. Amrouch, J. Martinez, M. Nafria, J. Henkel, "Designing Guardbands for Instantaneous Aging Effects," in DAC, 2016 (accepted for publication).

[3] J. Keane and C. H. Kim, "Transistor aging," *IEEE Spectrum*, 2011.

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

[5] "BSIM Compact MOSFET Models for SPICE Simulation," http://www-device.eecs.berkeley.edu/bsim/?page=BSIM4

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

[7] J. B. Velamala, V. Ravi, and Y. Cao, "Failure diagnosis of asymmetric aging under NBTI," in *ICCAD*, 2011, pp. 428–433.

[8] "Nangate, Open Cell Library," http://www.nangate.com/

[9] D. Lorenz, G. Georgakos, and U. Schlichtmann, "Aging analysis of circuit timing considering NBTI and HCI," in *IOLTS*, 2009, pp. 3–8.

[10] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu *et al.*, "The impact of NBTI on the performance of combinational and sequential circuits," in *DAC*, 2007, pp. 364–369.

[11] S. Kiamehr, F. Firouzi, and M. B. Tahoori. "Aging-aware timing analysis considering combined effects of NBTI and PBTI," in *ISQED*, 2013, pp. 53-59

[12] D. Gnad, M. Shafique, F. Kriebel, S. Rehman, D. Sun, and J. Henkel, "Hayat: Harnessing dark silicon and variability for aging deceleration and balancing," in *DAC*, 2015.

[13] N. Karimi, A. K. Kanuparthi, X. Wang, O. Sinanoglu, and R. Karri, "MAGIC: Malicious aging in circuits/cores," *ACM Trans. Archit. Code Optim.*, vol. 12, no. 1, 2015.

[14] M. Ebrahimi, F. Oboril, S. Kiamehr, and M. B. Tahoori, "Aging-aware logic synthesis," in *ICCAD*, 2013, pp. 61–68.

[15] H. Amrouch, V. M. van Santen, T. Ebi, V. Wenzel, and J. Henkel, "Towards interdependencies of aging mechanisms," in *ICCAD*, 2014, pp. 478–485.

[16] A. Kerber and T. Nigam, "Challenges in the characterization and modeling of BTI induced variability in metal gate / High-k CMOS technologies," in *IRPS*, 2013.

[17] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *Electron Devices, IEEE Transactions on*, vol. 53, no. 11, pp. 2816–2823, Nov 2006. http://ptm.asu.edu/

[18] http://iwls.org/iwls2005/benchmarks.html

[19] J. Henkel, L. Bauer, J. Becker, O. Bringmann, U. Brinkschulte, S. Chakraborty *et al.*, "Design and architectures for dependable embedded systems," in *CODES+ISSS*, 2011, pp. 69–78.

[20] http://trace.eas.asu.edu/yuv/index.html