Towards Aging-Induced Approximations

Hussam Amrouch*, Behnam Khaleghi†, Andreas Gerstlauer‡ and Jörg Henkel*
*Karlsruhe Institute of Technology, Chair for Embedded Systems (CES), 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*—In recent technology nodes, wide guardbands are needed to overcome reliability degradations due to aging. Such guardbands manifest as reduced efficiency and performance. Existing approaches to reduce guardbands trade off aging impact for increased circuit overhead. By contrast, the goal of this work is to *completely remove guardbands through exploring, for the first time, application of approximate computing principles in the context of aging*. As a result of naively narrowing or removing guardbands, timing errors start to appear as transistors age. We demonstrate that even in circuits that may tolerate errors, aging can be catastrophic due to unacceptable quality loss. Furthermore, quantifying such aging-induced quality loss necessitates expensive (often infeasible) gate-level simulations of the complete design. We show how nondeterministic aging-induced timing errors can be converted into deterministic and controlled approximations instead. We first translate the required guardband over time into an equivalent reduction in precision for individual RTL components. We then demonstrate how, based on pre-characterization of RTL components, we can quantify aging-induced approximation at the whole microarchitecture level without the need for further gate-level simulations. Results show that a 3 bit reduction in precision is sufficient to sustain 10 years of operation under worst-case aging in the context of an image processing circuit. This corresponds to an acceptable PSNR reduction of merely 8 dB, while at the same time increasing area and energy efficiency by 13%.

I. Introduction

Due to aging effects, the cost of sustaining reliable operation of circuits for their projected lifetime is continuously growing, and it is typically paid in terms of performance [1], [2]. Aging shifts the electrical characteristics of nMOS and pMOS transistors at operation time. The primary observation is an increase in threshold voltage (V_{th}). In turn, ΔV_{th} slows down transistors (due to the reduction in their drain current) and hence prolongs the delay of logic gates. Over time, aging makes the circuit's critical path (CP) larger, as clarified in Eq. 1 which employs a first-order approximation based on [3]. Therefore, to keep aging effects at bay, including a timing guardband (t_{GB}) [1], [2], [4] that corresponds to the maximum delay increase that can be caused by aging during the projected lifetime is indispensable.

$$
\begin{aligned}
&t_{CP} \leq t_{clock} \Rightarrow \textit{no timing errors} \checkmark \qquad (1)\\
&t_{CP} = \sum_{M_i \in CP} t_{M_i} \; ; \; t_{M_i} \propto \frac{1}{(V_{dd} - V_{th} - \Delta V_{th})^2}\\
&t_{clock}(Aging) = t_{CP}(noAging) + t_{GB}\\
&f_{clock}(Aging) < f_{clock}(noAging) \Rightarrow \textit{performance loss } \mathbf{!}
\end{aligned}
$$

As shown, including a timing guardband to overcome aging leads to clocking the design with a lower frequency than its potential. Hence, a performance loss is incurred, which might not be tolerated. Several approaches for narrowing guardbands and hence increasing performance have been proposed [2], [4]. However, existing approaches reduce aging impact at the cost of other circuit overhead like transistor sizing [5].

Approximate computing has recently emerged as a new design philosophy challenging the notion that inherently error-tolerant applications, such as multimedia systems always need to produce an exact result [6]. Existing approaches deliberately introduce errors and losses in output quality in exchange for other design-time aspects. In this work, we aim to explore, for the first time, opportunities for applying approximate computing concepts to *trade off guardbands for aging-induced quality degradations over time*.

For circuits in which errors can be tolerated, if the impact on quality due to narrowing or removing required guardbands can be accurately quantified, designers can have the freedom to selectively decide *when* (i.e. at which point of lifetime), *where* (i.e. in which RTL component) and *how much of* a guardband is employed. Quantifying aging-induced errors is, however, challenging. Errors due to aging originate from timing violations when the delay of gates increases. These timing violations are nondeterministic in the sense that the effect of aging on gate delays and the triggering of timing violations on different circuit paths depend on the complete history of inputs over lifetime and the applied inputs, respectively. Determining where and when errors originate as well as how they impact overall circuit output requires extremely time-consuming, if not infeasible, gate-level simulations of a design under aging and for given stimuli. This is, in fact, necessary to determine which paths within the netlist will be subject to timing violations under aging effects. Furthermore, aging affects all paths and non-uniformly increases their delay over time. As such, the whole design's netlist needs to be analyzed and timing violations can appear in any path, leading to potentially catastrophic errors or unbounded and intolerable quality loss. *Hence, quantifying aging-induced timing errors may not be feasible due to complexity and it may not be feasible to provide boundaries of such errors.*

In this work, we propose to narrow or remove guardbands by converting nondeterministic aging-induced timing errors into deterministic, controlled and bounded approximations instead. The most common and effective deterministic approximation strategy to operate typical applications under reduced timing budgets is to reduce the precision of basic arithmetic computations [7], [8]. This minimizes quality loss and allows providing upper bounds on error magnitude. We apply such approaches in the context of aging. In the process, we develop an effective method to quantify required aging-induced approximations and their quality impact without the need for time-consuming gate-level simulations of the complete design.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DAC '17, June 18-22, 2017, Austin, TX, USA
© 2017 ACM. ISBN 978-1-4503-4927-7/17/06...$15.00
DOI: http://dx.doi.org/10.1145/3061639.3062331

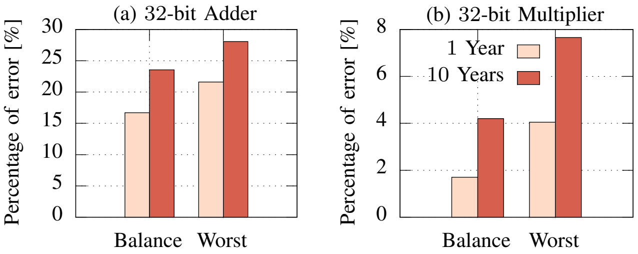


Fig. 1. Evaluating the impact of aging-induced errors on the correctness of adder and multiplier components under *balance* and *worst* stresses of aging.

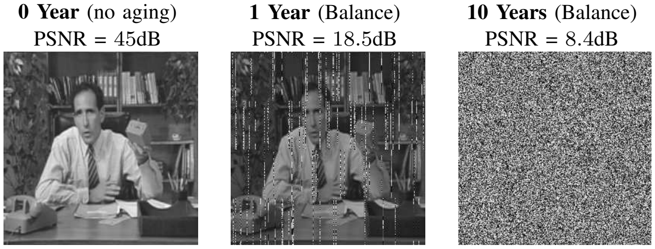


Fig. 2. Evaluation of the impact of aging-induced errors on image quality.

Our novel contributions within this paper are as follows: (1) To allow removing guardbands while sustaining reliability, we demonstrate how aging-related delay increases over time can be converted into an equivalent reduction in precision using a methodology that first pre-characterizes individual RTL components and then determines required approximations at the whole micro-architecture level without simulations.
(2) With this methodology, we study and quantify the impact of aging-induced errors and approximations on quality loss over time under removed or narrowed guardbands. Results show that, while timing errors lead to unacceptable quality losses already after 1 year, worst-case aging can be fully compensated for 10 years by approximations requiring only small reductions in precision and quality while simultaneously providing a 13% increase in area and energy efficiency.

II. Motivational Case Study

In the following, we study whether *do we really need to include a guardband even in error-tolerant circuits?* We first investigate the impact of aging-induced timing errors at the RTL-component level. Here, we study 32-bit adder and multiplier components. We then investigate how severe the quality drop at the microarchitecture level is due to aging-induced timing errors. We perform these studies for Discrete Cosine Transform (DCT) and Inverse Discrete Cosine Transform (IDCT) circuits as typically employed in multimedia designs to encode and decode images, respectively. For our analysis, we first obtained the maximum frequency in the absence of aging as reported by the synthesis tool. Each circuit (i.e. Adder, Multiplier, DCT and IDCT) has been synthesized under the highest optimization effort to achieve the best performance using the "*ultra_compile*" option from Synopsys. We then perform a static timing analysis (STA) of the synthesized netlist under aging. We use the recently proposed, publicly available aging-aware cell library [4], [9] for this purpose. The resulting standard delay file (.sdf) is finally used to perform gate-level simulations of the analyzed circuit under aging running at its maximum frequency determined in the absence of aging. This allows us to quantify how aging-induced delay increase in logic gates can result in errors in the circuit's output due to timing violations. We apply 10^6 values following a normal distribution as inputs to the adder and multiplier. In case of the DCT and IDCT, we first encoded and then decoded a representative image input from the "video trace library" [10] typically used in multimedia evaluations. For a wider investigation, we considered a lifetime of 1 to 10 years to demonstrate how aging-induced errors increase over time. Furthermore, we also considered worst (i.e. 100% stress) and balanced (50% stress) cases of aging. An analysis under worst case of aging is more conservative and provides an upper bound on aging impact. By contrast, the analysis under the balance case provides us with a more typical aging impact.

Fig. 1 summarizes our results. The percentage of errors in the adder and multiplier outputs reaches 20% and 4% after one year, respectively, and it increases to 28% and 8%, respectively, after 10 years of worst-case aging. As can be noticed, the impact of aging can be quite different from one RTL component to another. Fig. 2 presents the output image of the DCT-IDCT chain along with corresponding PSNR. As shown, aging-induced timing errors lead to an unacceptable drop in quality after *just one year* and a noisy/useless image after 10 years, respectively. The probability of error in the IDCT output reaches 15% and 100% after 1 year and 10 years. This results in a drop in the quality (w.r.t. PSNR) to 18.5dB after 1 year and 8dB after 10 years. Note that 30dB is commonly considered an acceptable image quality [11].

In summary, aging-induced timing violations lead to a high probability of error and to an unacceptable quality drop. Therefore, removing guardbands to retain efficiency is not possible without controlling aging-induced errors and hence converting them from nondeterministic to deterministic ones. Achieving this goal through employing approximate computing principles represents concisely the core of this paper.

III. Related Work

Several approaches have been aimed at reducing the required guardband to overcome aging. For instance, [12] captures the most susceptible paths at design time and then forces the synthesis tool to employ stronger gates/cells along these paths. The work in [5] proposed an NBTI-aware transistor re-sizing under multiple operating conditions to compensate NBTI-induced delay increase while reducing the side-effect on power. Most recently, the work in [4] proposed aging-aware logic synthesis of circuits using a so-called degradation-aware cell library that contains the delay information of gates/cells under aging effects. An improved version after modeling the impact of aging on power was proposed in [13]. While these approaches reduce the cost of guardbanding w.r.t. performance, they all result in higher area, leakage and dynamic power. *Our work is orthogonal to such approaches. It enables trading off guardbands for precision instead of traditional design aspects.* The authors in [4] showed that their technique is able to sustain a high image quality even in the absence of a guardband. However, all existing techniques still require a guardband to guarantee timing correctness. Therefore, expensive gate-level simulations are necessary to evaluate the impact of timing violations under reduced or removed guardbands. As mentioned earlier, performing such gate-level simulations may be infeasible. For instance, a gate-level simulation of the DCT-IDCT chain, which consists of $\sim 2\times10^6$ gates, can take around 4 days for a single image with 1920×1080 resolution on an Intel quad-core Xeon CPU with 2.9 GHz and 4 GB of RAM.

A range of techniques have been studied for approximate computing at the hardware level [6]. Early work applied voltage overscaling to achieve energy savings, using algorithmic or architectural extensions to correct, mitigate or strategically accept associated timing errors [14]–[16]. However, none of the existing work has considered effects of timing errors due to aging. More recently, approximations through controlled logic simplifications in fundamental arithmetic building blocks, such as adders and multipliers have been explored [7], [8], [17]. This can improve both area and switching activity as well as reduce errors and delays, but, again, relationships with aging have not been investigated. A plethora of approximate adders and multipliers have been proposed in recent years [6]. *Our work is orthogonal to and allows applying any such component approximations.* Without loss of generality, we use precision reduction through truncation of least significant bits (LSBs) as generic approximation technique in this paper.

IV. Characterizing Aging-Induced Approximations at the RTL Component Level

As explained in Section I and Eq. 1, aging increases the delay of logic gates and hence leads to longer maximum delays and critical paths of combinational RTL components. Therefore, the initial clock period, $t_{clock}(noAging)$ determined at design time in the absence of aging can not be sustained later in a design's lifetime. However, when we reduce the precision of a component, its initial delay reduces as the critical path becomes potentially shorter. As such, the aging-induced delay (i.e. the required guardbands) and the total delay after aging also become smaller. We can therefore characterize individual combinational RTL components to relate a decrease in their aging-induced and total delays to an equivalent reduction in precision. From this, we create a library of such RTL components pre-characterized for aging. Following standard approximation requirements, we perform such pre-characterization for all datapath components, where we assume that control blocks/stages are separated and protected against aging through traditional means, such as stronger gates.

Estimating aging-induced delay increase: Due to recovery and healing effects in aging, the total number of defects, which defines the final impact of aging on the transistor's delay – represented by ΔV_{th} as shown in Eq. 1 – is determined by the *stress factor* (S), which corresponds to the ratio between how long the transistor was under stress and how long it was under recovery. In general, estimating the impact of aging can be done either under *worst-* or *actual-*case aging as follows:

Worst-case aging: Every transistor is under the highest aging stress in which $S = 100\%$. Hence, the maximum ΔV_{th} will be induced. Even through this analysis is definitely conservative (if not unrealistic), it is in fact necessary to provide us with the maximum possible delay increase due to aging independent of the activities under which the RTL component will later be employed. Note that protecting the circuit against worst-case aging provides a guarantee that no timing errors due to aging will ever occur during the projected lifetime.

Actual-case aging: The switching activities caused by inputs applied to the RTL component are considered to determine the *actual* S per transistor and hence the actual ΔV_{th}. This analysis provides us with the delay increase due to aging under a certain set of inputs. Since this analysis is dependent on the input data, we consider two cases in our work: 1) input data following a *normal distribution*, which aims to be independent of a specific application scenario, and 2) input data extracted from a specific running application. Note that we selected a normal distribution as being representative for typical image processing data. However, our proposed characterization is not limited to a specific kind of data distribution.

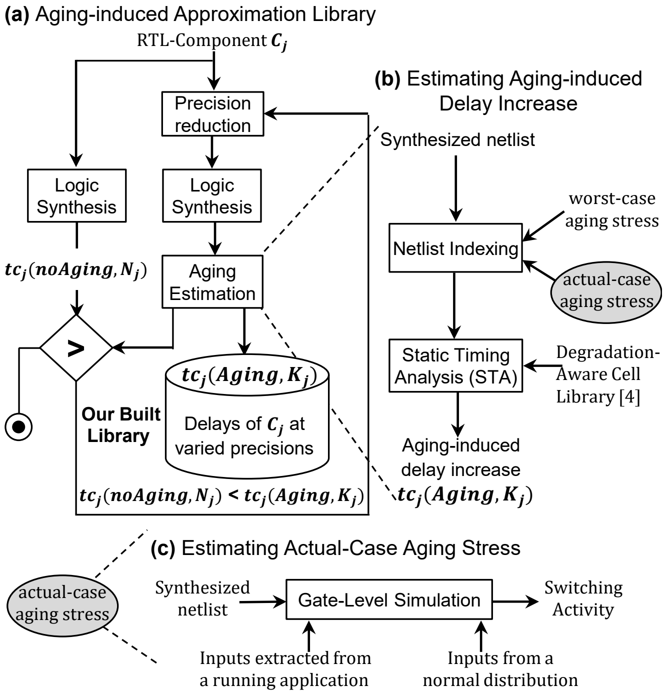

(a) Aging-induced Approximation Library
RTL-Component C_j
Precision reduction
Logic Synthesis
Logic Synthesis
$tc_j(noAging, N_j)$
Aging Estimation
>
$tc_j(Aging, K_j)$
Delays of C_j at varied precisions
Our Built Library
$tc_j(noAging, N_j) < tc_j(Aging, K_j)$
(b) Estimating Aging-induced Delay Increase
Synthesized netlist
Netlist Indexing
worst-case aging stress
actual-case aging stress
Static Timing Analysis (STA)
Degradation-Aware Cell Library [4]
Aging-induced delay increase
$tc_j(Aging, K_j)$
(c) Estimating Actual-Case Aging Stress
actual-case aging stress
Synthesized netlist
Gate-Level Simulation
Switching Activity
Inputs extracted from a running application
Inputs from a normal distribution

Fig. 3. Characterization of aging-induced component approximations linking the delay of components under aging to an equivalent reduction in precision.

To estimate the aging-induced delay increase in an RTL component, we first synthesize it using the original (i.e. degradation-unaware) cell library. This provides us with maximum delay in the absence of aging. Such delay, in practice, is the timing constraint that must be fulfilled during lifetime despite aging effects. Then, we annotate the netlist according to the targeted case of aging stress. The annotation can be done either evenly when worst-case aging is targeted (i.e. all transistors are indexed with 100%), or, when actual-case aging is targeted, by annotating the netlist with the averaged estimated S caused by the switching activities due to the applied input set. Afterwards, we perform STA of the annotated netlist along with the aforementioned degradation-aware cell library [4] which is publicly available at [9]. This library contains the delay information of logic gates under (11×11) different stress factors of pMOS/nMOS transistors.

Creating library of aging-induced approximations: For a given RTL component C_j with a bit-width of N_j, we can iteratively reduce its precision until we reach a specific point ($K_j < N_j$) at which the component's delay under aging is able to fulfill the initial critical path delay, as Eq. 2 demonstrates:

$$t_{C_j}(Aging, K_j) \leq t_{C_j}(noAging, N_j) : K_j < N_j \qquad (2)$$

$$\Rightarrow \textit{uncontrolled timing errors} \rightarrow \textit{controlled approximation} \checkmark$$

$$\Rightarrow \textit{no added guardband} \rightarrow \textit{no efficiency loss} \checkmark$$

$t_{C_j}(Aging, K_j)$ and $t_{C_j}(noAging, N_j)$ are the aged and unaged delays of component C_j at reduced precision K_j and full precision N_j, respectively. Fig. 3 demonstrates the required steps of our proposed characterization. Fig. 3(a) shows the general flow for creating our proposed library of aging-induced approximations. Fig. 3(b) shows how the aging-induced delay increase is estimated either under the worst- or actual-case of

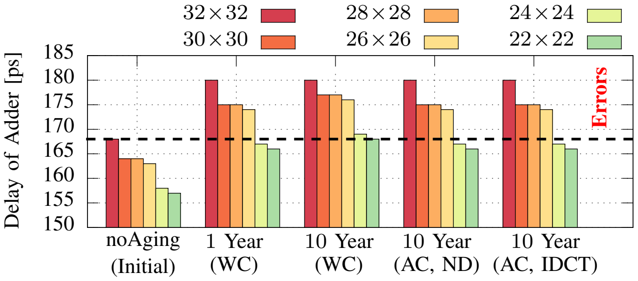


Fig. 4. Characterization of a 32×32 bit adder to convert aging-induced timing errors into an equivalent precision reduction. WC: *worst-case* aging. (AC, ND): *actual-case* aging under inputs from a *normal distribution*. (AC, IDCT): *actual-case* aging under IDCT inputs of an image.

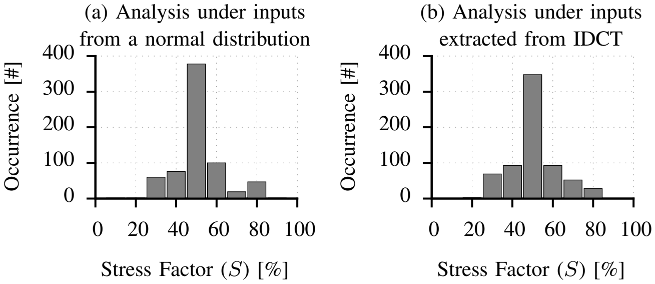


Fig. 5. Stress factors under actual-case aging considering inputs from a normal distribution and from the IDCT resulting in very similar stress distributions.

aging. The latter necessitates gate-level simulations as clarified in Fig. 3(c). Note that gate-level simulations to characterize RTL components under actual-case aging are a *one-time* effort with much smaller complexity than at the full microarchitecture level. For instance, full characterization of our multiplier and adder took less than an hour. The aforementioned characterization to convert the aging-induced delay increase into an equivalent precision reduction is performed offline for different individual RTL components in order to build a *library of aging-induced approximations*. Such a library enables us later, when we study a whole microarchitecture, to determine for every existing RTL component the required reduction in its precision to meet required timing constraints.

Fig. 4 shows an example of our characterization of a 32×32 bit adder. We consider both worst- and actual-case aging. For the latter, we use stimuli from a normal distribution as well as inputs extracted from an IDCT decoding a sample image. As shown, reducing the precision enables us to compensate the aging-induced delay increase. For instance, reducing the precision by merely 2 bits allows us to *narrow* the required guardband by 31%. Reducing the precision further down to 24 bits will be sufficient to *completely remove* the guardband until 1 year of worst-case aging. Precision needs to be reduced down to 22 bits for a reliable operation (i.e. without any aging-induced timing errors) over 10 years without adding any guardband. However, considering actual-case (instead of worst-case) aging provides us with a less conservative precision reduction of only 8 bits. Note, however, that considering actual-case of aging will not provide a guarantee that timing errors will never occur during the projected lifetime.

Sufficiency of considering normal distribution: From the analysis in Fig. 3 we observe that, when performing the characterization under the actual-case aging, considering normally distributed inputs versus inputs extracted from the application provides identical results w.r.t. precision reduction. This is because the corresponding aging-induced delay is very similar. Such similarity is due to similar stress factors distributions between different input stimuli. This is confirmed by the histograms of S shown in Fig. 5. Both histograms are similar and hence the induced delay increase in gates will be similar as well. As such, it is generally sufficient to employ artificial inputs for characterization. We repeated this analysis for a 32×32 multiplier showing similar conclusions.

All in all, the required characterization of an RTL component to convert aging-induced timing errors (under the actual-case) into approximations can be performed independent from the running application. Alternatively, worst-case aging can be considered. Hence, it is guaranteed that the applied approximation prevents any aging-induced timing error independent from the usage of the component.

V. Quantifying Aging-Induced Approximations at the RTL Microarchitecture Level

In this section, we demonstrate how aging-induced delay increases at the component level can be converted into approximations for a whole microarchitecture design. In the following, we explain our proposed process step by step, and we summarize all steps in Fig. 6. Note that our approach is general and does not depend on a specific approximation technique, as long as the technique allows trading off between precision and delay. In this paper, we applied our approach to integer arithmetic as most of the approximation techniques in literature have been explored in this space. However, as long as techniques are available that satisfy the condition of trading off accuracy for delay, our approach can be equally applied to other circuits, such as floating point arithmetic.

Obtaining timing constraints: We first synthesize the design to obtain its critical path (CP) delay in the absence of aging ($t_{CP}(noAging)$). This represents the required timing constraint that the whole design must fulfill during its projected lifetime (e.g., 10 years) and under the targeted aging stress condition (e.g., worst-case aging).

Aging estimation: We then preform an aging-aware STA for the whole design to obtain the delay of every combinational datapath block B_k within the RTL design under aging ($t_{B_k}(Aging)$). This allows us to calculate the available timing slack $t_{B_k}(Slack) = t_{CP}(noAging) - t_{B_k}(Aging)$ between the timing constraint and the delay under aging for every register-transfer block. This timing slack tells us whether timing violations may occur as this block ages during lifetime.

Our selective approximation: A *negative* time slack (i.e. $t_{B_k}(Slack) < 0$) means that timing violations will occur in the corresponding block. Hence, we need to convert aging-induced timing errors into approximations. To achieve that, we employ our pre-built library (see Section IV) to select the required precision reduction with which the existing time slack is compensated. We assume that every block B_k contains *one* RTL database component C_j. In practice, glue and steering logic also exist. However, a reduction in the precision of C_j will result in a proportional decrease in complexity and delay of such glue logic after synthesis. As such, we use the relative slack $t_{B_k}(relSlack) = t_{B_k}(Slack)/t_{CP}(noAging)$ of block B_k to determine the precision $K_j \leq P_j < N_j$ that will achieve the same relative delay reduction $t_{C_j}(Aging, P_j) \leq (1 + t_{B_k}(relSlack)) \times t_{C_j}(noAging, N_j)$ in component C_j. Depending on how large the existing time slack is, the precision reduction can be either up to K_j (i.e. the maximum precision reduction to fully compensate aging) or smaller. When the time

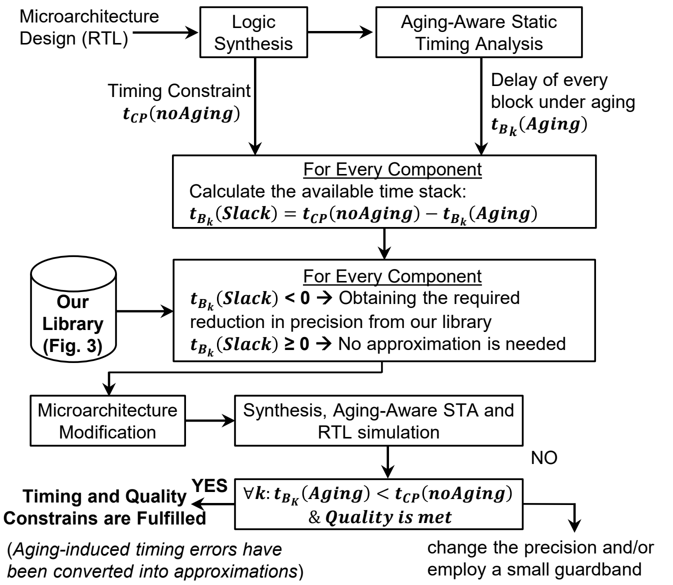

Microarchitecture Design (RTL)
Logic Synthesis
Aging-Aware Static Timing Analysis
Timing Constraint $t_{CP}(noAging)$
Delay of every block under aging $t_{B_k}(Aging)$
For Every Component
Calculate the available time stack:
$t_{B_k}(Slack) = t_{CP}(noAging) - t_{B_k}(Aging)$
Our Library (Fig. 3)
For Every Component
$t_{B_k}(Slack) < 0$ → Obtaining the required reduction in precision from our library
$t_{B_k}(Slack) \geq 0$ → No approximation is needed
Microarchitecture Modification
Synthesis, Aging-Aware STA and RTL simulation
NO
YES
Timing and Quality Constrains are Fulfilled
$\forall k: t_{B_k}(Aging) < t_{CP}(noAging)$ & $Quality$ is met
(Aging-induced timing errors have been converted into approximations)
change the precision and/or employ a small guardband

Fig. 6. Overview of applying aging-induced approximations to an RTL design.

slack is *positive* (i.e. $t_{B_k}(Slack) \geq 0$), there is no need for any approximations as aging will not cause any timing violations. Hence, the component stays at its full precision (i.e. $P_j = N_j$). *Therefore, in practice, the different RTL components within a microarchitecture design may have different reductions in their precision or may not be approximated at all.*

Validation: After determining the required P_j for every component, we modify the RTL design accordingly. We then synthesize the design and perform an aging-aware STA and a functional RTL simulation to ensure that the timing and quality constraints are met. Note that there is a very small likelihood – even though we had not observed such a case in our evaluation – that small negative timing slacks remain. This can be due to either aging-induced delay increases or less-than-estimated impact of precision reductions in the glue logic surrounding RTL components. In such a case, changing the precision and/or adding a small timing guardband to compensate slack will be necessary. However, such a guardband will be significantly smaller than the original case when no approximations are applied. If final quality is not sufficient, precision can be increased and a resulting guardband be similarly added.

VI. Evaluation and Comparisons

We first present results about aging-induced approximations at the RTL component level. Then, we demonstrate our methodology of converting guardbands into aging-induced approximations at the whole microarchitecture level. We employ Synopsys Design Compiler for synthesis along with its STA tool to calculate maximum delay of a circuit. The power analysis has been also done using Synopsys tool after taking the switching activities induced by the simulated input stimuli into account. As mentioned before, to estimate the impact of aging, we employed the publicly available, degradation-aware cell libraries from [4], [9] which are based on the 45nm open-source NanGate library [18]. These libraries are compatible with the Synopsys tool flows and contain the delay information of various cells under different aging stress conditions. During synthesis, we use the “ultra compile” Synopsys option to achieve the best possible optimization w.r.t. performance. We employ Mentor ModelSim as RTL and gate-level simulator. Note that gate-level simulations are only necessary to estimate switching activities for one-time component characterization under actual-case aging.

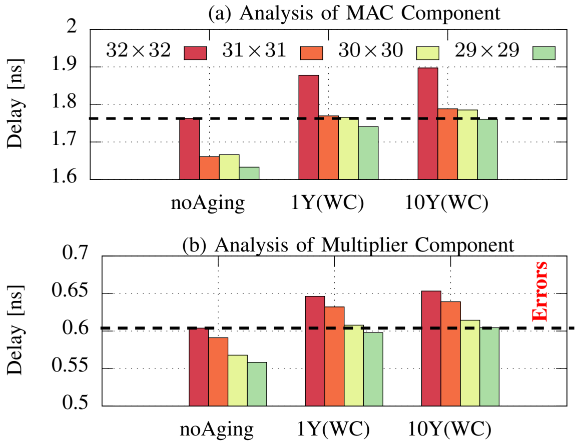


Fig. 7. Characterizing MAC and Multiplier components demonstrating how aging-induced delay increases can be converted into precision reductions.

RTL component level: We target three different RTL components: an adder, a multiplier and a multiply-accumulate (MAC) unit. We selected a width of 32 bits as base precision for each example. We characterize each component under the worst-case (WC) aging for a lifetime of 1 and 10 years as explained in Section IV. To reduce precision, we use truncation of LSBs as approximation method in our experiments. Note, however, that our methodology is general and supports application of arbitrary component approximations. Fig. 7 shows characterization results for the multiplier and MAC. Adder results have been previously presented in Section IV. For the multiplier and MAC, reducing the precision by only 1 bit results in narrowing the guardband by 29% and 80%, respectively, after 10 years of worst-case aging stress. Reducing the multiplier precision by 2 bits further narrows its 10-year guardband down to 79%. Finally, reducing the precision of multiplier and MAC by 2 and 3 bits is sufficient to fully compensate aging after 1 and 10 year(s) of worst-case stress, respectively. This shows that there is a large design freedom to trade off the employed guardband for a gradual precision and hence quality reduction. Note that in the case of the adder, a larger precision reduction (6 and 8 bits for 1 and 10 years, respectively) was necessary as shown in Fig. 4. This demonstrates how different RTL components necessitate different reductions in precision over time when aging-induced timing errors are converted into approximations.

RTL microarchitecture level: We use the IDCT from Section II as RTL design example. We apply our proposed methodology in Section V to convert the aging guardband into approximations. In the context of our studied IDCT, the multiplier is the component in the critical path, which has a negative relative timing slack of -8.3% after worst-case aging for 10 years. Results from our library and final validation show that it is sufficient to reduce the precision of the multiplier by 3 bits to fully compensate for such aging effects. Other RTL components have enough timing slack to remain at their full precision without needing any approximations. In Fig. 8(a), we show a comparison of IDCT delays between the aging-unaware (i.e. original) design and our aging-induced approximations. As can be seen, after converting aging-induced delay increases into approximations, our design fulfills the required timing constraint in all aging cases. Therefore, no timing errors will occur during operation and only our induced approximations will contribute to the

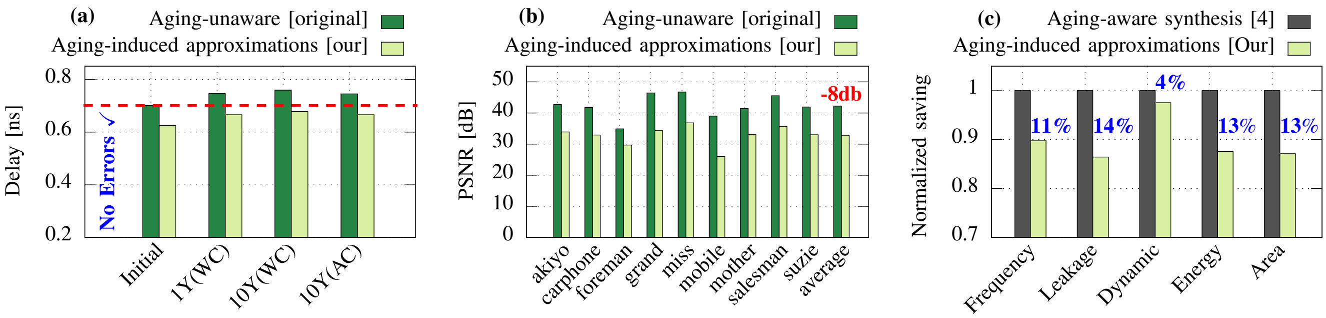
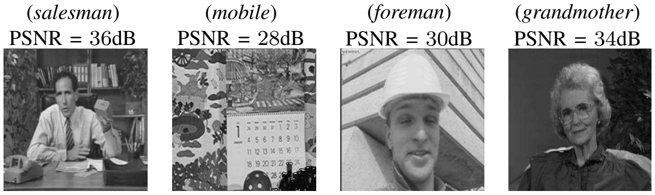


Fig. 8. (a) Comparison between IDCT delays for an aging-unaware design and our aging-induced approximations determined, confirming that our design fulfills the required timing constraint. (b) Quality of various images when our aging-induced approximations are applied for worst-case aging and a lifetime of 10 years. (c) Achieved savings after applying aging-induced approximations normalized to results of state-of-the-art aging-aware synthesis from [4].


(*salesman*) PSNR = 36dB

(*mobile*) PSNR = 28dB

(*foreman*) PSNR = 30dB

(*grandmother*) PSNR = 34dB

Fig. 9. Examples of images when applying our aging-induced approximations after 10 years of worst-case aging.

drop in image quality. Fig. 8(b) shows the PSNR (which is a typical metric used to represent image quality) for various images decoded by an IDCT when our aging-induced approximations are implemented. PSNR was obtained from RTL simulations, which required a few seconds each on our quad-core 2.9GHz Xeon machine. Even for an image with 1920×1080 resolution, the required RTL simulation takes less than 3 minutes. Note that quantifying aging-induced timing errors (instead of approximations) using gate-level simulations would need 4 days. The examined images for this analysis were obtained from the "video trace library" [10], which is a standard benchmark for multimedia evaluations. A lifetime of 10 years under worst-case aging was used to determine approximations and thus present the lower boundary of PSNR. PSNR on average merely drops by 8db and it is above 30db in all images except one ("mobile"), where it is slightly below. Examples of obtained images are presented in Fig. 9. As noticed, even for the "mobile" image with 28db PSNR, image quality is still very good and noise is hardly observable.

We further compare our results against the state-of-the-art approach from [4]. We already employ their released degradation-aware cell libraries in our work. The work in [4] proposed an aging-aware logic synthesis approach in which their cell libraries are employed to synthesize a circuit netlist more resilient against aging. Fig. 8(c) presents our achieved savings in terms of area, performance, power and energy compared to their aging-aware logic synthesis. Due to the removed guardband, our design is 11% faster. In addition, using aging-induced approximations results in 14% and 4% less leakage and dynamic power consumption, respectively. This translates into 13% energy savings. This is because when we reduce precision, logic complexity is reduced and fewer logic gates will be required (13% area saving). Thus, both leakage power and switching activity are reduced. As such, converting guardbands into approximations not only improves performance, but, instead of incurring overhead, allows us to inherit additional approximate computing benefits.

VII. Summary and Conclusions

In this paper, we investigated how approximate computing principles can be applied towards increasing the efficiency of circuits while overcoming aging. We demonstrated how aging-induced delay increases resulting in nondeterministic and catastrophic timing errors can be compensated by deterministic and controlled reductions in precision. We further provide an effective approach to determine required precision reductions and quantify resulting quality losses without time-consuming (if not infeasible) gate-level simulations. This is a paradigm shift providing designers a new degree of freedom to narrow or remove guardbands in exchange for degradations in quality, which, instead of increasing, reduce area and power overhead. *Hence, precision in error-tolerant circuits can be traded off with performance and efficiency while sustaining reliability.* By applying approximations adaptively we can envision future systems that gradually degrade in quality as they age over time.

Acknowledgments

This work is partially supported by a Humboldt Research Fellowship and the German Research Foundation (DFG) priority program "Dependable Embedded Systems" (SPP 1500). The authors thank Ku He for his valuable support.

References

[1] J. Keane and C. H. Kim, "Transistor aging," *IEEE Spectrum*, 2011.
[2] S. Arasu, M. Nourani, J. M. Carulli, and V. K. Reddy, "Controlling aging in timing-critical paths," *IEEE D&T*, vol. 33, no. 4, pp. 82–91, 2016.
[3] "BSIM Compact MOSFET Models for SPICE Simulation," http://www-device.eecs.berkeley.edu/bsim/?page=BSIM4.
[4] H. Amrouch, B. Khaleghi, A. Gerstlauer, and J. Henkel, "Reliability-aware design to suppress aging," in *DAC*, 2016.
[5] S. Roy, D. Liu, J. Singh, J. Um, and D. Z. Pan, "OSFA: a new paradigm of aging aware gate-sizing for power/performance optimizations under multiple operating conditions," *IEEE TCAD*, vol. 35, no. 10, pp. 1618–1629, 2016.
[6] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *ETS*, 2013.
[7] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: Imprecise adders for low-power approximate computing," in *ISLPED*, 2011.
[8] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," in *ICCAD*, 2012.
[9] "Degradation-Aware Cell Libraries, V1.0," http://ces.itec.kit.edu/dependable-hardware.php
[10] "Video trace library," http://trace.eas.asu.edu/yuv/index.html.
[11] N. Thomos, N. Boulgouris, and M. Strintzis, "Optimized transmission of JPEG2000 streams over wireless channels," *IEEE TIP*, vol. 15, no. 1, pp. 54–67, 2006.
[12] M. Ebrahimi, F. Oboril, S. Kiamehr, and M. B. Tahoori, "Aging-aware logic synthesis," in *ICCAD*, 2013, pp. 61–68.
[13] H. Amrouch, S. Mishra, van Santen Victor, M. Souvik, and J. Henkel, "Impact of bti on dynamic and static power: From the physical to circuit level," in *IRPS*, 2017.
[14] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE TVLSI*, vol. 9, no. 6, p. 813823, 2001.
[15] G. Karakonstantis, D. Mohapatra, and K. Roy, "System level DSP synthesis using voltage overscaling, unequal error protection & adaptive quality tuning," in *SIPS*, 2009.
[16] K. He, A. Gerstlauer, and M. Orshansky, "Controlled timing-error acceptance for low energy IDCT design," in *DATE*, 2011.
[17] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," 2008.
[18] "Nangate, Open Cell Library," http://www.nangate.com/.