

Synthesis and Characterization of Approximate Circuits to Mitigate the Aging and Temperature Effects in an Advanced CMOS Technology

by

FRANCISCO JAVIER HERNANDEZ SANTIAGO

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science
in
Computer Engineering

Department of Electrical and Computer Engineering

University of Alberta

© Francisco Javier Hernandez Santiago, 2020

Abstract

While the goal has been increasing performance and reducing power consumption by decreasing the transistor size, the most advanced semiconductor technologies (i.e., those with dimensions smaller than 45 nm) have become more susceptible to high temperatures and aging phenomena. As a consequence, the circuit performance (i.e., speed) is degraded significantly over time, which may lead to timing violations in the critical path delays. Designers commonly add timing margins to a circuit as guard-bands to guarantee circuit reliability during its projected lifetime. However, guard-banding significantly affects circuit performance. Many design methodologies have been investigated to mitigate the effect of guardbands, but these methodologies come at the cost of other circuit overhead, such as an extra margin in the transistor size or voltage. In contrast, the principle of approximate computing has emerged as a promising solution to improve circuit measures (including speed, power and area) by intentionally introducing controllable errors in resilient applications. Hence, our main objective is to explore how approximate arithmetic circuits can be employed to deal with circuit degradations without a loss in performance.

The methodology presented in this thesis consists of converting degradations to

deterministic and controllable errors, rather than using guard-bands to guarantee reliability. That is, an approximate circuit is characterized by considering its degradations to obtain the same performance of an accurate circuit without any degradation. Our simulation results show that the simple use of truncated arithmetic circuits leads to a higher quality loss compared to using other approximate circuits. To guarantee the same performance, for instance, Cartesian genetic programming-generated adders and lower-part OR gate-based adder result in the lowest mean relative error distances (MREDs) among all the considered approximate adders, independently of the number of years (from 1 to 10 years) or the level of temperature (from 25 °C to 70 °C) the circuit is supposed to be reliably operating. A truncated multiplier has the lowest MRED towards a reliable operation in 10 years, but the approximate multipliers with configurable error recovery are most suitable when the level of degradation is higher, e.g. at a temperature of 70 °C. For three different image processing applications, our conducted experiments show that guard-bands can be mitigated while maintaining an output result with good visual quality.

Keywords

Approximate computing; approximate adders; approximate multipliers; low-power circuits; high-performance circuits; semiconductor device reliability; hardware accelerators.

Preface

Chapter 3 was partially published in the GLSVLSI'19, Proceedings of the 29th IEEE/ACM Great Lakes Symposium on VLSI, Washington, D.C., USA, 2019 as “Characterizing Approximate Adders and Multipliers Optimized under Different Design Constraints.” Honglan Jiang, Francisco J. H. Santiago, Mohammad Saeed Ansari, Leibo Liu, Bruce F. Cockburn, Fabrizio Lombardi and Jie Han. I developed the Verilog code for some circuits, performed synthesis and ran the Monte Carlo simulations to obtain the circuit measures. I also was in charge of presenting the article at the conference. Dr. S. Ansari provided the code for some designs. Dr. H. Jiang developed the VHDL code for most of the approximate circuits and wrote the article. Dr. L. Liu, Dr. B. Cockburn and Dr. F. Lombardi provided comments and suggestions for the manuscript. Dr. J. Han supervised this work and revised the manuscript.

Acknowledgments

I owe my deepest gratitude to my entire family. Thank you for being close and providing support through all these years that I have been away from home. There is no more I could ask for.

I want to express my gratitude to my research advisor Dr. Jie Han for his continuous support during my studies. Thanks for allowing me to work on a project that I knew very little at the start, but have learned so much from. I also greatly appreciate for introducing me with great researchers, especially to Honglan Jiang. I have been amazingly fortunate to have a collaboration with her.

Several institutions make this research work possible. Thanks to CONACYT and FUNED for their financial support that allowed me to cover my expenses and tuition in Canada. Thanks to the University of Alberta and the Queen's University for sponsor my trips to conferences and a training course. Finally, last but not least, thanks to the Canadian Microelectronics Corporation for giving me access to EDA tools and providing technical support during my research.

Table of Contents

Abstract	ii
Preface	iv
Acknowledgments	v
Table of Contents	vi
List of Tables	ix
List of Figures	x
List of Abbreviations	xiv
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Thesis Contributions	5
1.3 Outline	6
Chapter 2: Background	8
2.1 Sources of Transistor Variations and Degradations	8
2.1.1 Manufacturing Process Variations	9
2.1.2 Supply-Voltage Variations	10
2.1.3 Aging Effects	11
2.1.4 Temperature	12
2.2 Techniques for Coping with Degradations	13
2.2.1 Design-time Synthesis	14
2.2.2 Adaptive Techniques	15
2.2.3 Approximate Computing	17
2.3 Summary	19

Chapter 3:	Characterizing Approximate Arithmetic Circuits under Aging and Temperature Effects	21
3.1	Review of Approximate Circuits	22
3.1.1	Manual Designs	22
3.1.2	Automated Design	26
3.1.3	Error Metrics	27
3.2	Characterization Under Different Design Constraints	28
3.2.1	Experimental Setup	29
3.2.2	Evaluation of Approximate Adders	29
3.2.3	Evaluation of Approximate Multipliers	32
3.3	Estimating Error Metrics Under Aging-Induced Delay	36
3.3.1	Experimental Setup	36
3.3.2	Evaluation of Approximate Adders	37
3.3.3	Evaluation of Approximate Multipliers	38
3.4	Characterizing Delay Guard-bands	39
3.4.1	Simulation Results	39
3.5	Summary	42
Chapter 4:	Trading-off degradations for approximations at the Component Level	43
4.1	Design Methodology	44
4.2	Towards Aging-Induced Approximation	47
4.2.1	Experimental Setup	47
4.2.2	Simulation Results	47
4.3	Towards Temperature-Induced Approximation	50
4.3.1	Experimental Setup	50
4.3.2	Simulation Results	52
4.4	Summary	55
Chapter 5:	Trading-off degradations for approximations at the Architectural Level	57
5.1	Image Processing Applications	58
5.1.1	DCT/IDCT	58
5.1.2	Image Smoothing	59
5.1.3	Image Sharpening	59
5.2	Motivational Study Case	60
5.3	Design Methodology	62
5.4	Experimental Setup	65
5.5	Simulation Results	66
5.6	Summary	70

Chapter 6: Conclusions	72
6.1 Summary	72
Bibliography	75

List of Tables

3.1	Summary of the approximate adders.	32
3.2	Summary of the unsigned approximate multipliers.	35
4.1	Characterizing 16-bit high-performance approximate circuits towards aging-induced approximation. The circuits are ranked by the mean squared error (MSE) from the lowest to the highest value.	48
4.2	Characterizing 16-bit low-power approximate circuits towards aging-induced approximation. The circuits are ranked by the MSE from the lowest to the highest value.	51
4.3	Characterizing 16-bit high-performance approximate circuits towards temperature-induced approximation. The circuits are ranked by the MSE from the lowest to the highest value.	52
4.4	Characterizing 16-bit low-power approximate circuits towards temperature induced approximation. The circuits are ranked by the MSE from the lowest to the highest value.	54
5.1	Measures for the high-performance applications running at $70^{\circ}C$. . .	67
5.2	Measures for the low-power applications running at $70^{\circ}C$	68

List of Figures

1.1	Main design methodologies that have improved the performance of computing systems in the last years (adapted from [6]).	2
1.2	Source of errors in error resilient applications (adapted from [7]).	3
3.1	Building a $2n \times 2n$ multiplier using four $n \times n$ multipliers. The principle is illustrated for $n = 8$	25
3.2	A circuit comparison of the 16-bit approximate adders synthesized for high-performance. The parameter k for LOA and TruA ranges from 2 to 9, for ESA and ACA from 8 down to 3 (except 7), for CSA from 5 down to 3, and for the remaining adders from 6 down to 3, all from right to left in the figures. Regarding the CGPAs, the configurations with the lowest error metrics for a specific PDP are reported.	30
3.3	A circuit comparison of the 16-bit approximate adders synthesized for low-power. The parameter k for LOA and TruA ranges from 2 to 9, for ESA and ACA from 8 down to 3 (except 7), for CSA from 5 down to 3, and for the remaining adders from 6 down to 3, all from right to left in the figures. Regarding the CGPAs, the configurations with the lowest error metrics for a specific PDP are reported.	31

3.4	A circuit comparison of the 16-bit approximate multipliers synthesized for high-performance. The number of truncated LSBs for TruM, TBM, and BBM varies from 1 to 7 from right to left. The number of MSBs used for error compensation varies from 16 down to 10 for AM1, AM2, TAM1, and TAM2 from right to left. The mode number for ACM is from 4 to 3 from right to left. Regarding PPAM and CGPMs, the best designs in terms of hardware metrics and low-error are reported. . . .	33
3.5	A circuit comparison of the 16-bit approximate multipliers synthesized for low-power. The number of truncated LSBs for TruM, TBM, and BBM is from 1 to 7 from right to left. The number of MSBs used for error compensation is from 16 down to 10 for AM1, AM2, TAM1, and TAM2 from right to left. The mode number for ACM is from 4 to 3 from right to left. Regarding PPAM and CGPMs, the best designs in terms of hardware metrics and low-error are reported.	34
3.6	Characterization of different errors in the 16-bit approximate adders' output due to timing violations. The parameter k for LOA and TruA ranges from 2 to 9, for ESA and ACA from 8 down to 3 (except 7), CSA from 5 down to 3 and for the remaining adders from 6 down to 3, all from right to left. Regarding CGPAs, the configurations with the lowest error metrics for a specific delay are reported.	37

3.7 Characterization of different errors in the 16-bit approximate multipliers' output due to timing violations. The number of truncated LSBs for TruM, TBM, and BBM varies from 1 to 7 from right to left. The number of MSBs used for error compensation varies from 16 down to 10 for AM1, AM2, TAM1, and TAM2 from right to left. The mode number for ACM varies from 4 to 3 from right to left. Regarding PPAM and CGPMs, the best designs in terms of hardware metrics and low-error are reported. 38

3.8 Characterizing delay-guardbands of 16-bit high-performance approximate adders under different aging and temperature effects using the proposed methodology in [31]. The parameter k for LOA and TruA ranges from 2 to 9, for ESA and ACA from 8 down to 3 (except 7), CSA from 5 down to 3 and for the remaining adders from 6 down to 3, all from right to left. Regarding the CGPAs, the configurations with the lowest error metrics for a specific delay are reported. 40

3.9 Characterizing delay-guardbands of 16-bit high-performance approximate multipliers under different aging and temperature effects using the proposed methodology in [31]. The number of truncated LSBs for TruM, TBM, and BBM is from 1 to 7 from right to left. The number of MSBs used for error compensation is from 16 down to 10 for AM1, AM2, TAM1, and TAM2 from right to left. The mode number for ACM is from 4 to 3 from right to left. Regarding PPAM and CGPMs, the best designs are reported. 41

4.1	Design methodology at the component level to convert degradations to controllable errors using approximate circuits.	45
5.1	Image quality output of three different image processing applications when the chip is working in nominal conditions ($25^{\circ}C$).	61
5.2	Image quality output of three different image processing applications after the circuit is exposed at $70^{\circ}C$ without a technique to overcome the transistor degradations.	61
5.3	Design methodology at the architectural level to convert degradations to controllable errors using approximate circuits.	63
5.4	IDCT outputs when the chip is exposed at $70^{\circ}C$ using: (a) degradation-aware synthesis with an accurate circuit, (b) our approximate approach with TBM-7, and (c) our approximate approach with BBM-8.	69
5.5	Image smoothing outputs when the chip is exposed at $70^{\circ}C$ using: (a) degradation-aware synthesis with an accurate circuit, (b) our approximate approach with TAM1-16, and (c) our approximate approach with TruM-7.	70
5.6	Image sharpening outputs when the chip is exposed at $70^{\circ}C$ using: (a) degradation-aware synthesis with an accurate circuit, (b) our approximate approach with TAM1-16, and (c) our approximate approach with TruM-7.	70

List of Abbreviations

ACA almost correct adder

ACAA accuracy-configurable approximate adder

ACM approximate compressor-based multiplier

AM1 approximate multiplier 1

AM2 approximate multiplier 2

ATPG automatic test pattern generation

BAM broken array multiplier

BBM broken Booth multiplier

BIST built-in self test

BTI bias temperature instability

CCA consistent carry approximate adder

CGP Cartesian genetic programming

CGPAs Cartesian genetic programming-generated adders

CGPMs Cartesian genetic programming-generated multipliers

CLA carry-lookahead adder

CP critical path

CSA carry skip adder

CSPA carry speculative adder

DCT discrete cosine transform

ED error distance

EDA electronic design automation

ER error rate

ESA equal segmentation adder

ETAII error-tolerant adder type II

ETM error-tolerant multiplier

FA full adder

GCSA generate signals-exploited carry speculation adder

HCID hot carrier induced degradation

HDL hardware description language

ICM inaccurate counter-based multiplier

IDCT inverse discrete cosine transform

ILD inter-layer dielectric

LOA lower-part OR gate-based adder

LSBs least significant bits

MED mean error distance

MRED mean relative error distance

MSBs most significant bits

MSE mean squared error

NBTI negative bias temperature instability

NMED normalized mean error distance

NMOS negative channel metal-oxide semiconductor

PBTI positive bias temperature instability

PDP power-delay product

PMOS positive channel metal-oxide semiconductor

PPAM product perforation-based approximate multiplier

PSNR peak signal-to-noise ratio

PVT process, voltage and temperature variations

RCA ripple-carry adder

RED relative error distance

RTL register transfer level

SCSA speculative carry selection adder

SDF standard delay file

SoC System-on-Chip

STA static timing analysis

TAM1 truncated approximate multiplier 1

TAM2 truncated approximate multiplier 2

TBM truncated Booth multiplier

TDDB time-dependent dielectric breakdown

TDP thermal design power

TruA truncated adder

TruM truncated multiplier

UDM under-designed multiplier

Chapter 1

Introduction

1.1 Motivation

The number of electronic devices connected to the Internet is projected to increase to one trillion worldwide by 2025 [1]. All of these devices in the form of sensors will generate a massive quantity of data that requires more advanced computing systems for efficient processing. As a projection, an increase of 1000x in performance is expected to meet the computing requirement by 2030 [2]. Unfortunately, increasing performance has become the most challenging problem for hardware designers in the past years. One might think that the continuous scaling of the transistor would continuously improve the circuit's performance. However, the continued miniaturization of transistors has been presenting limited gains in performance and energy efficiency. Moreover, reaching the nanometer scale feature size has endangered the reliability of the entire system due to variations in the manufacturing process, fluctuating temperatures and aging phenomena.

Complex and often conflicting design constraints are required to optimize power,

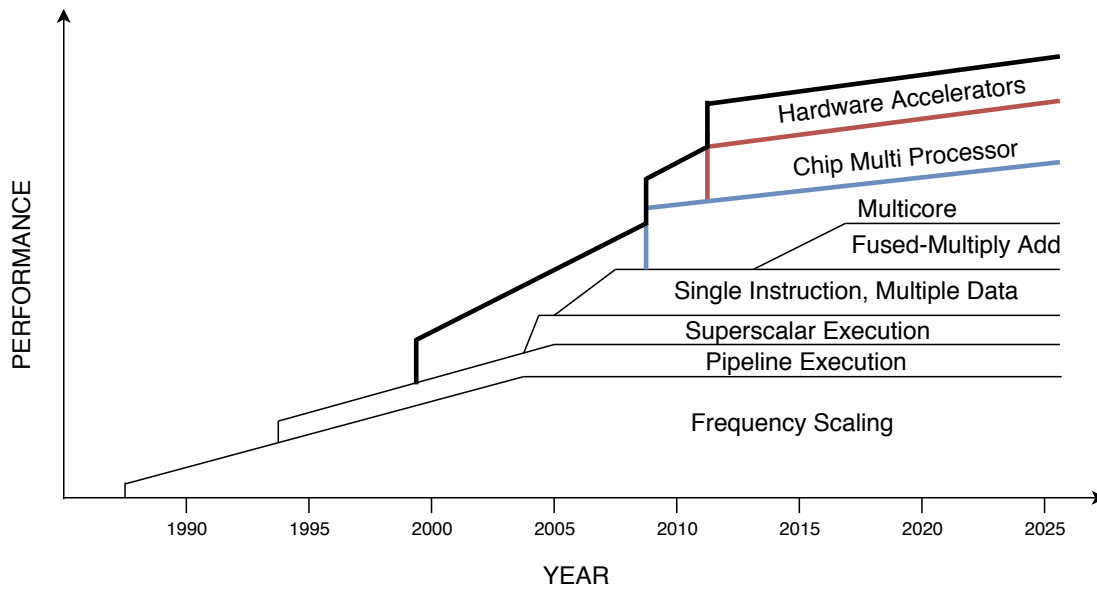


Figure 1.1: Main design methodologies that have improved the performance of computing systems in the last years (adapted from [6]).

performance, and reliability. While an increasing number of smaller transistors improves performance if the power density remains constant (known as Dennard scaling), it also significantly increases the leakage current, which in turn leads to thermal challenges [3]. For instance, transistors become notably slow as temperature increases from the nominal value considered during design (25 °C) to the worst-case value that may occur in operation (70 °C) [4]. The difficulty and cost of cooling the devices during the peak power have required a temporal and spatial shutdown of on-chip resources, which is also known as the dark-silicon problem [5]. Other sophisticated design methodologies, such as pipelining execution, heterogeneous architectures, multiprocessors, and hardware accelerators, have also been investigated to improve the performance (see Fig. 1.1) [6]. However, the rapid growth of data would still demand more energy-efficient computing techniques for modern applications.

The avalanche of real-time data has also generated different workload profiles. An

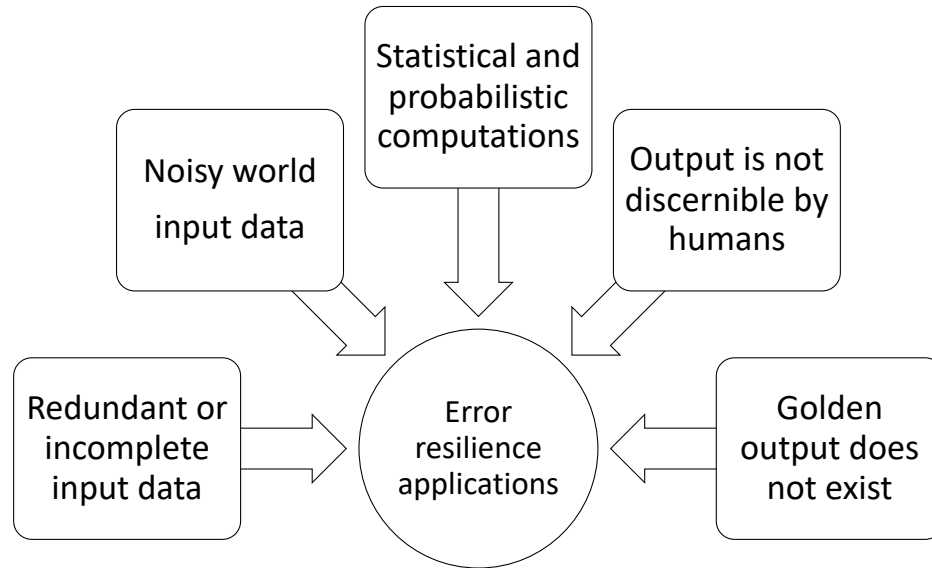


Figure 1.2: Source of errors in error resilient applications (adapted from [7]).

essential characteristic of many emerging workloads is their error tolerance, which is broadly defined as the property of one system to continue normal operations if one component fails. Although computation errors are not desirable, multiple factors (see Fig. 1.2) allow the use of this methodology in modern applications [7]. Firstly, image processing, recognition, computer vision, and artificial intelligence applications operate under statistical and probabilistic computations. Second, often the input data is noisy, and for most of the time, redundant. Moreover, a golden output does not exist or the difference between accurate and approximate results is simply not discernible by humans. A recent study shows how modern applications employed 83% of their run-time in resilient kernels [7]. Therefore, this promising approach, commonly called approximate computing, can be used as a solution to improve hardware metrics at the cost of relaxing the requirement of exact numeric computations for the next generations of integrated circuits [8, 9].

Recent years have witnessed tremendous progress in the field of approximate computing at different levels of abstraction. From a hardware perspective, multiple techniques have been proposed to introduce errors at the system level such as voltage-over-scaling or over-clocking [10, 11]. Similarly, various arithmetic circuits in which the component does not precisely perform the functionality of an accurate circuit have been investigated at the transistor, gate, and architectural level [12, 13]. Different from conventional arithmetic operations, stochastic computing is another promising approach that uses bit-wise operations to process probabilities [14]. At the software layer, accuracy can be traded for speed and energy efficiency using compilers, programming languages, or modifying the algorithms to skip resilient tasks [15, 16, 17].

Based on the above observations, approximate computing can be considered as a potential solution to meeting the processing requirements while saving hardware resources in comparison to accurate computations. However, this methodology has not considered the decline of technology scaling, especially certain aspects that jeopardize the correct functionality of chips in the nano-CMOS era [18]. Among the multiple factors that affect circuit reliability, temperature and aging are at the forefront. These phenomena degrade the circuit performance over time, which may lead to timing violations in the critical path delay. Consequently, controllable errors coming solely from the approximate circuits are transformed into catastrophic errors, even for resilient applications, if the degradations are not carefully considered during the design stage. Therefore, this thesis investigates the challenge of increasing hardware performance while sustaining reliability using the principles of approximate computing. By doing so, the impact of aging and temperature fluctuations is effectively mitigated in an advanced technology node.

1.2 Thesis Contributions

In this thesis, we present an exhaustive evaluation of approximate arithmetic circuits under different workload conditions, an analytical framework to accurately trade-off degradations for approximations at the component level, and a comparative evaluation of degradation-induced approximation at the architectural level. In particular, the novel contributions of this work are as follows:

1. The basics and recent developments of approximate arithmetic circuits are first reviewed for evaluation purposes. Our analysis used application-specific metrics such as the mean squared error (MSE) during the evaluation. The hardware description language (HDL) code of the state-of-the-art approximate arithmetic circuits is developed to optimize the quality of results during the synthesis process. Finally, timing, area, and power optimizations are achieved using advanced algorithms using the Synopsys Design Compiler tool.
2. For the first time, the approximate adders and multipliers are evaluated in terms of their reliability under different levels of degradations, e.g., aging and temperature scenarios. Gate-level simulations are performed to quantify the output quality towards timing-induced violations. To avoid non-deterministic errors due to transistor degradations, a state-of-the-art guardband technique is employed for comparison.
3. Recent work has shown how approximations can be traded off for delay guardbands, but it did so exclusively for truncation of the least significant bits (LSBs). This thesis explores the applicability of different approximation techniques for such a methodology with the purpose of trading-off guardbands for a minimum

quality loss in the output. To be competitive with the state-of-the-art in terms of the clock frequency, the accurate designs from Synopsys DesignWare library are used as a comparison benchmark.

4. We demonstrate how by bringing in reliability-aware design at earlier stages (at the component level), the circuit's lifetime reliability at the architectural level is guaranteed while still meeting the customer requirements. In this process, design space exploration is also enabled without the need for running time-consuming simulations for applications that demand different qualities. Finally, the effectiveness and limitations of the error metrics at the application level (e.g., peak signal-to-noise ratio (PSNR)) are investigated in comparison with the error metrics obtained at the component level.

1.3 Outline

Five chapters follow this introductory chapter. Chapter 2 discusses why integrated circuits have become more susceptible to various kinds of degradations such as process, voltage and temperature variations, and aging phenomena. Then, the most recent work is reviewed to address these problems efficiently, including approximate computing. Chapter 3 presents an evaluative comparison of approximate circuits performance variation under different level of stresses of aging and temperature. In Chapter 4, arithmetic circuits are characterized to link guard-bands to an equivalent reduction in precision in order to sustain reliability. Chapter 5 discusses how by bringing degradation-aware techniques at earlier stages, we could optimize the architectures for different hardware requirements and applications. Finally, chapter

6 summarizes the main contributions of this thesis.

Chapter 2

Background

Multiple factors affect the switching speed of transistors, and thus the timing of logic circuits. This chapter presents the background of integrated circuit performance variation, and the most effective design methodologies that have been proposed previously to improve circuit reliability.

2.1 Sources of Transistor Variations and Degrada- tions

Since the prediction of Gordon Moore in 1965, the scaling of transistors has brought significant gains in performance and power [19]. However, transistors have also become more susceptible to various kinds of degradations, which affect the reliability of a circuit over its lifetime. The sources of degradations that generate performance variability and thus jeopardize the correct operation of an integrated circuit have been mainly process, voltage and temperature variations (PVT). Later on, the aging

phenomena have also become a real concern in terms of reliability once the semiconductor manufacturing process reached the nano-meter feature size (i.e., ≤ 45 nm) [20].

2.1.1 Manufacturing Process Variations

The manufacturing process has always been one of the most prominent variation concerns in the design of integrated circuits. During the process of manufacturing, the characteristics of fabricated integrated circuit hardly match the nominal values obtained during the design stage. In addition, the performance may differ from one integrated circuit to another, which has led to a significant yield loss. Historically, the chip yield—a quality metric of a manufacturing process—has reduced from 90% to almost 50% at 90nm, and around 30% in the 45nm feature size [21]. Unfortunately, the trend will become more pronounced with the continuous shrinking of transistors. Despite the systematic variations (e.g., limitations in the lithography process) that can be easily predicted, random variations are of particular attention for their complexity in the nano-CMOS era [22].

The leading causes of random variations in integrated circuits are classified into two categories: (1) variations in transistor parameters, and (2) variations in the interconnect structure. Variations in transistor parameters occur mainly from variations in the channel length, channel width, channel doping, or variations of the gate oxide thickness [23]. These types of variations directly affect the performance and power characteristics of transistors. In fact, experimental results have shown that the performance difference between dies in the same wafer can be up to 30%, while the differences in the leakage current can vary by up to a factor of 20 [22]. For

this reason, circuits are designed with a slow corner case for the negative channel metal-oxide semiconductor (NMOS) and positive channel metal-oxide semiconductor (PMOS), i.e., the transistors are characterized by the lowest supply voltage combined with the highest possible temperature. While the probability for this worst-case scenario to occur is almost null, guaranteeing operation of these worst-case conditions ensures reliability for every condition the end-user could experience.

On the other hand, variations in the interconnect structure refer to differences in the structure of layers. Nowadays, the most complex integrated circuits are composed of 15 or more overlapping layers. The interconnection structure is composed of the different materials (typically *Al* and *Cu*), and adjacent levels of interconnect are separated from each other by an inter-layer dielectric (ILD). The dimensions of the ILD thickness, including line-spacing and metal-thickness, have become the primary sources of performance variation in the structure of a circuit. Fluctuations in the values of these dimensions affect the resistance and capacitance directly in an integrated circuit, which in turn, result in a performance loss [24].

2.1.2 Supply-Voltage Variations

The origin of voltage variation is mainly caused by the IR drop (also known as voltage drop) and di/dt noise. While IR drop is due to the current flow over the parasitic resistance of the electrical grid, di/dt noise is caused by the parasitic inductance in combination with the resistance of the power grid and package [25]. These effects, often called power noise effects, can lead to either voltage drops or voltage over shoots. A higher voltage increases the switching speed of transistors due to the corresponding increase in the current. By contrast, lower voltages result in lower speeds. However,

a lower voltage is of a particular concern as propagation delay drops in the range of nanoseconds to microseconds can lead to timing faults in the logic circuit if the critical path delay is activated during that moment [22].

2.1.3 Aging Effects

Historically, semiconductor manufactures used to employ the same scaling factor for the supply voltage (S_V) and transistor length (S_L) to increase performance and maintain the electric field constant. Unfortunately, the simple scaling of transistor feature size appears to have broken down. This occurs because the supply voltage can not be scaled down anymore, otherwise it would fall below the threshold voltage. With the continuous search to increase circuit performance, the rule of scaling in semiconductor fabrication changes from ($S_V = S_L$) to ($S_L < S_V$) [18]. As a result, the electric field across the channel and gate has been increasing in the most recent semiconductor technologies. With a higher electric field, the aging phenomena become stronger, which in turn, may break the gate dielectric. This phenomenon, called time-dependent dielectric breakdown (TDDB) may result in the total failure of the circuit. To solve this problem, semiconductor manufacturers replaced the material employed to build the gate dielectric layer with a more resistant material (i.e. high-K dielectric materials) [26]. Despite the probability that a TDDB occurs has decreased, the employment of such a new materials impacts the transistor characteristics due to other aging phenomena [18].

The most prominent degradations in transistors due to the employment of high-K materials have been categorized as bias temperature instability (BTI), and hot carrier induced degradation (HCID). The negative bias temperature instability (NBTI) and

positive bias temperature instability (PBTI) are two different forms of BTI. NBTI degrades PMOS transistors while PBTI degrades NMOS transistors. On the other hand, HCID degrades both types of transistors. BTI and HCID were reported since 1966, but they only became a significant issue once the gate oxide thickness scaled to values lower than $1.5nm$. As mentioned, a high electric field is the key source behind aging induced degradations. Carriers, which are accelerated by the electric field, collide with the gate rather than moving between the drain and the source when a PMOS/NMOS transistor is in operation. The collision that occurs degrades transistors due to the charges that get trapped inside the dielectric. While a vertical electric field over the gate stimulates the BTI phenomenon, a lateral electric field across the channel stimulates the HCID phenomenon. Both phenomena degrade significantly the carrier mobility and threshold voltage in the transistors, thus reducing performance over the transistors' lifetime [18].

2.1.4 Temperature

Temperature fluctuations are caused by elevated ambient temperature or heat due to power dissipation in a chip. Temperature imposes a design constraint called thermal design power (TDP) [5]. TDP constraint defines the maximum amount of heat that the cooling system can dissipate. Since an integrated circuit does not perform any mechanical work, most of the consumed energy is dissipated in thermal energy. Once the temperature starts rising beyond the limits, the transistor becomes slower due to reduced carrier mobility and interconnect resistance, which in turn leads to timing violations [22].

2.2 Techniques for Coping with Degradations

The sources of variability in integrated circuits are either independent (manufacturing process) or dependent (voltage, temperature, and aging phenomenon) on integrated circuit lifetime, which increases the complexity of the design of efficient and reliable computer systems. In this thesis, we focus primarily on time-dependent variations. Therefore, to avoid timing violations due to circuit performance variations, safety guard-bands are commonly added on top of the nominal values.

Three different types of guardbands have been used in the past years: timing, voltage, and gate-sizing. As transistors become slow due to the aforementioned degradations, designers can increase the clock period in such a way that the critical path delay would be smaller than the clock period during the expected circuit's lifetime. Unfortunately, this leads to a significant loss in performance. On the other hand, a voltage guardband can also be added to the top of the nominal voltage to increase the transistor's current, and thus switching speed. This approach allows us to run the circuit at the highest performance, but it will also consume more power. Both timing and voltage guardbands can be adjusted during run-time. However, the third type of guardband, gate-sizing, is a permanent fix and consists of designing stronger gates during the manufacturing process, which increases the chip area and, consequently, power consumption. Moreover, larger circuits result in fewer dies per wafer, which also increases the manufacturing cost.

The main drawback of timing, voltage, and gate-sizing is that they require analysis of the worst-case scenarios. However, worst-case conditions are almost impossible to define. On the one hand, it is unknown if the transistor's parameters will be shifted during the process of synthesis, layout, or integration of the whole System-on-Chip

(SoC). On the other hand, it has been recently demonstrated that the worst-case degradation uniformly applied to each transistor does not capture the actual worst-case of a cell [27]. Therefore, the circuits are typically over-designed with pessimistic guard-bands. Based on these observations, more sophisticated techniques have been investigated to mitigate the impact of guardbands in integrated circuits — for example, design-time synthesis, adaptive techniques and, most recently, approximate computing.

2.2.1 Design-time Synthesis

Bringing transistors degradation to industry electronic design automation (EDA) tools is essential to guarantee reliability in integrated circuits. Industrial tools employ a large variety of techniques such as re-sizing logic gates, register re-timing optimization, multiple clocks, clock-gating, and automatic re-partitioning algorithms to optimize integrated circuits [28]. All of these techniques, including dynamic voltage and frequency scaling, can alleviate the effects of performance variations due to degradations. For this reason, degradation-aware synthesis methodologies have been investigated in the past years to improve circuits' reliability.

Based on the observation that logic cells aged differently, an aging-aware logic synthesis methodology is proposed in [29]. This approach improves circuits reliability with stronger gates in the critical paths that are more vulnerable to aging effects. This approach also releases constraints on paths with a smaller post-aging delay to compensate for the impact on the area. Multiple iterations of re-synthesis are required to find an optimal solution. Experimental results indicate that a high level

of reliability can be obtained without area overhead. A similar aging aware gate-sizing methodology is introduced in [30]. Different from [29], this approach not only considers NBTI to optimize area with stronger gates, but also consider the supply voltage for power optimization under different work conditions.

The work presented in [31], instead of using stronger gates, employs delay guard-bands. To optimize the aging guard-bands with static timing analysis (STA), the authors developed 121 degradation-aware cell libraries, where each library contains a different workload scenario. These libraries can also be employed during the process of synthesis. By this means, the effect of aging is also mitigated using aggressive and efficient algorithms from the Synopsys Design Compiler tool. The authors demonstrated that ignoring carrier mobility leads to overestimating guard-bands by almost 20%. This work is consequently improved in [32] to characterize the impact of aging on dynamic and static power. The same authors also proposed static and adaptive optimization techniques for temperature guard-bands in [4]. Both have been evaluated in five well-known processors, and experimental results show that delay guard-bands can be reduced by 22% compared with traditional approaches.

2.2.2 Adaptive Techniques

With the increasing demand for high-speed circuits, a loss in performance when guard-bands are not required, is not acceptable. This fact has increased interest in adaptive techniques that maintain performance while ensuring reliability. These techniques mainly consist of adjusting safety margins as needed with Dynamic Voltage and Frequency Scaling (DVFS) in real-time.

Zhang *et al.* proposed a *schedule voltage scaling* for NBTI minimization and compensation [33]. This technique adopts a time scheduler to increase the operating voltage gradually instead of operating at a fixed voltage guardband during the circuit's lifetime. In the design, the authors investigate the impact on the number of voltage levels. Final results indicate that using ten voltage levels in the scheduler enhances circuit lifetime reliability by 46% using a 45 nm technology process.

Das *et al.* proposed a voltage management design technique for error detection and correction called Razor-II [34]. This novel design replays operations in the processors when a soft error occurs after the voltage and frequency are scaled. While error detection occurs at the flip-flop level, the subsequent correction is performed at the architectural level. This technique takes advantage of high-performance microprocessors that support speculative operations, such as out-of-order execution and branch prediction. Razor-II was implemented and validated in the Breazeale Nuclear Reactor at the University of Pennsylvania to quantify the soft error rate. The experiments conducted show an improvement of 33% in energy-efficiency under the worst-case scenarios.

More sophisticated adaptive techniques employing machine learning algorithms have been investigated. Sadi *et al.* present a new framework for designing SoCs with self-adaption capability against aging-induced degradation. This methodology uses built-in self test (BIST) to monitor critical paths in the design. Therefore, the primary step in this framework was the introduction of automatic test pattern generation (ATPG) targeting the high-usage critical paths under aging-induced delay. The BIST mechanism feeds a machine learning algorithm with the results to predict the aging degradation state. Predicted results are used to activate a remedy against

timing degradation [35].

2.2.3 Approximate Computing

Design-time synthesis and adaptive techniques in real-time have improved hardware resilience against performance variation [31, 29, 30, 32, 33, 34, 35, 36]. Unfortunately, all these conventional techniques increase the overall chip cost significantly due to a higher area, power, delay, and time-consuming simulations. By contrast, approximate computing has emerged recently as a solution to improve reliability due to degradations without adding unnecessary hardware overhead. It can be used in combination with conventional techniques to mitigate degradation effects by trading off output quality instead of adding delay or voltage guard-bands.

Software Level

Palomino *et al.* addressed the temperature problem by varying the degree of approximation in video coding applications [37]. The proposed technique classifies the resilience of different video regions to control the level of approximation in the workload. Different modes of approximation, selected at run-time, are applied depending on the Quality of service (QoS) of the application. The larger the level of approximation, the lower the executed workload, and thus less thermal dissipation will result. Experimental results show that this technique can reduce the temperature by $10^{\circ}C$ while still maintaining good visual quality results compared to state-of-the-art techniques.

Design-time Synthesis

Most recently, approximate computing has been employed to mitigate guard-bands at the circuit level. Amrouch *et al.* proposed *aging-induced approximation* [38]. First, the authors confirmed that aging-induced timing errors lead to an unacceptable quality drop even for inherently error-tolerant applications. To address this problem, instead of using delay guard-bands to sustain reliability, the authors converted the aging-induced timing errors into controllable and deterministic errors coming solely from the arithmetic computations. Experimental results show that truncation of 10 and 3 least significant bits (LSBs) is enough to sustain reliability in a 32-bit adder and multiplier, respectively. In the context of an image processing application, this methodology not only eliminated delay guard-bands at the architectural level but also enhances energy efficiency by 13% with a PSNR higher than 30 dB.

Adaptive Techniques

Boroujerdian *et al.* proposed two approaches for synthesizing delay-configurable circuits to overcome temperature variations and narrow guard-bands [39]. These configurable circuits minimize quality losses by dynamically and adaptively applying quality scaling in the presence of temporary circuit degradations. This approach also takes advantage of automatic re-partitioning algorithms from the EDA tools. The first approach consists of duplicating approximate arithmetic circuits to overcome different levels of temperature without sharing any resources among them. This approach accurately sets the narrowest guardbands for each possible scenario, but suffers from significant energy and an area overhead. The second approach decreases the energy

and area overhead by sharing the resources among the circuits. Selection of the appropriate approximate circuits is based on the measurement of temperature in real-time. Results of an IDCT application show up to a 21% speedup with a PSNR higher than 39 dB in the image output.

An adaptive technique, called *aging gracefully approximation*, was presented in [40]. This design measures the effects of aging using a state-of-the-art monitoring system in real-time. The monitor consists mainly of three blocks: (1) a control block, (2) a delay chain, and (3) an encode block. The first propagates a signal through the delay chain, which is the critical path in a circuit. A sampler counts how many elements are active during the signal propagation, and the result is encoded to 5-bit delay output information. If the result does not meet the timing requirement, the proposed design excludes the computation in the LSBs instead of fixing the values of voltage or clock frequency. By this means, the designers guarantee that the clock period is large enough to perform the computations in the most significant bits (MSBs). Experimental results, compared to conventional guard-bands approaches, show an improvement of 19.8% and 10.2% for dynamic and static power, respectively.

2.3 Summary

This chapter discussed how technology nodes reached an inflection point where continuous scaling increases the susceptibility of integrated circuits to various kinds of failures. Among the most important aspects that affect circuits reliability, temperature and aging effects are the forefront. This problem, including the end of Dennard scaling, makes the design of reliable and efficient computer systems difficult. Although a large number of solutions have been proposed to improve hardware efficiency and

prolong chip lifetime, most of them suffer from energy and delay overhead. On the other hand, approximate computing has emerged as another solution to mitigate degradations. However, only the truncation of LSBs has been investigated so far to overcome degradations. Considering that multiple approximate circuits have been proposed in the past years, a comparative study needs to be performed in order to determine if there are more effective approximation schemes with better trade-offs among error, power, and speed.

Chapter 3

Characterizing Approximate Arithmetic Circuits under Aging and Temperature Effects

In this chapter, approximate arithmetic circuits are reviewed and classified according to two principal design methodologies. Next, the approximate circuits are characterized under different design constraints. We then demonstrate how approximate circuit performance varies if aging effects are not considered during the design stage. Finally, a state-of-the-art guardband technique is applied to guarantee reliability under aging and temperature effects.

3.1 Review of Approximate Circuits

Approximate circuits have been classified into two principal categories: traditional manually designed circuits and automatically designed circuits. Manual circuit designs employ hardware description languages to create a high-level representation of an approximate circuit, while the automated method employs advanced algorithms to iteratively modify a circuit by removing logic gates until the customer requirements are met [41].

3.1.1 Manual Designs

Approximate Adders

A common method to gain hardware efficiency at the cost of errors is to use carry speculation for the approximate adders. Considering that the carry propagation usually does not cover the entire n -bit adder, the addition can be accomplished using smaller sub-adders working in parallel. Therefore, various methods have been investigated to appropriately select the carry-in of each sub-adder.

The equal segmentation adder (ESA) uses a straightforward technique that significantly reduces the critical path delay by just fixing the carry inputs of sub-adders to “1” or “0” [42]. Therefore, there is no carry propagation among sub-adders, which leads to a significant reduction in the critical path delay without a hardware overhead.

The error-tolerant adder type II (ETAII) introduces a sub-carry generator to speculate the carry input of each sub-adder [43]. The carry skip adder (CSA) also complements each sub-adder with a sub-carry generator but the propagate signals of the previous $(i - 1)^{th}$ sub-adder decide if the carry-in comes from the $(i - 1)^{th}$ or

$(i - 2)^{th}$ sub-carry generator [44]. The generate signals-exploited carry speculation adder (GCSA) differs from the CSA in the carry selection, i.e., the carry-in is selected by its own propagate signals rather than its previous block [45]. The carry speculative adder (CSPA) uses two carry generators (one with carry-0 and the other with carry-1) and one carry predictor for each sub-adder; the output of the previous carry predictor is used to select one carry generator [46].

In the speculative carry selection adder (SCSA), an n -bit adder is divided into $\lceil \frac{n}{k} \rceil$ blocks. Each block is made of two k -bit sub-adders. The difference between both sub-adders is the carry input (one with carry-0 and the other with carry-1). One of them is selected as the final result using a multiplexer, which is controlled by the carry-out of one of the previous sub-adders [47]. Similarly, the consistent carry approximate adder (CCA) also uses two sub-adders for each block, but the selection of one of them is not based only on the previous block, but also on the current block [48].

The n -bit almost correct adder (ACA) is composed of an array of n sub-adders of length k (where $k < n$). The critical path delay for this design is reduced to $\mathcal{O}(\log k)$ at the cost a significant area and power overhead [49].

In an n -bit accuracy-configurable approximate adder (ACAA), the carry chain is cut to reduce critical-path delay [50]. But, a sub-adder is introduced for each cut chain to increase the accuracy. Therefore, the ACAA consists of $(\frac{n}{k} - 1)$ $2k$ -bit sub-adders where each sub-adder adds $2k$ consecutive bits with an overlap of k -bits. Half of the most significant sum bits for each sub-adder is selected as the partial sum.

Other approaches with limited hardware gains consist of replacing the adders at the LSBs with approximate cells, while the MSBs are implemented in an accurate and

conventional adder topology. The so-called lower-part OR gate-based adder (LOA) replaces each full adder (FA) in the LSBs with a bit-wise OR operation [51]. A simple technique, resulting in a so-called truncated adder (TruA), consists of completely removing the logic gates in the LSBs.

Approximate Multipliers

The multiplication operation can be broken down into three different stages: (1) partial product generation, (2) partial product accumulation, and (3) a final addition. Research work in approximate multipliers commonly introduces errors in the partial product generation and/or partial product accumulation stage.

The truncated multiplier (TruM) removes the logic gates in the LSBs to significantly reduce the hardware in partial product generation and partial product accumulation. The broken array multiplier (BAM) eliminates carry-save adders in an array multiplier in both the horizontal (by the elimination of partial-product rows) and/or vertical (by the elimination of partial-product columns) directions [51].

Zervakis *et al.* introduce a product perforation-based approximate multiplier (PPAM) [52]. This technique omits the generation of partial products (not necessarily starting from the LSBs) and can be applied to different multiplier structures such as Dadda and Wallace trees.

Kulkarni *et al.* propose an under-designed multiplier (UDM) to introduce error exclusively when the partial products are generated with a Karnaugh-map [53]. To save one output bit, the 2×2 multiplication result of “1001” is simplified to “111” when the operands are both “11”. Consequently, a larger width multiplier can be built as shown in Fig. 3.1.

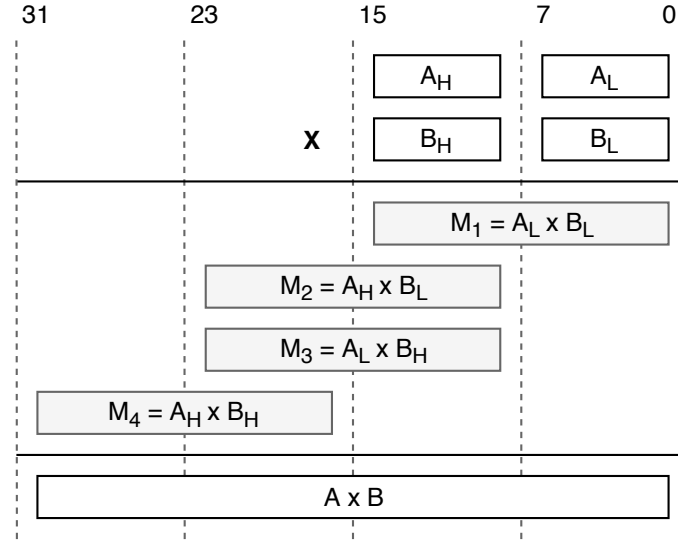


Figure 3.1: Building a $2n \times 2n$ multiplier using four $n \times n$ multipliers. The principle is illustrated for $n = 8$.

The so-called error-tolerant multiplier (ETM) splits the operands into two parts, which are not necessarily equal. The bits of the upper part are used to perform an accurate multiplication while the bits of the lower part are used to perform either an approximate or an accurate multiplication. A mechanism is introduced to decide which one is the most appropriate based on the output magnitude, i.e., high-magnitude outputs are computed accurately [54].

An approximation can also be obtained using approximate counters or compressors in the partial product tree. An inaccurate counter-based multiplier (ICM) is proposed in [55]. The multiplication is performed with an approximate (4:2) counter using a Wallace tree. The carry and sum in the approximate counter are approximated as “10” for “100” when all input signals are “1”. In [56], two approximate (4:2) compressor designs are proposed for a Dadda multiplier using 4 different schemes. This approximate multiplier is referred to as an approximate compressor-based multiplier

(ACM).

Liu *et al.* proposed four approximate multipliers with configurable error recovery in [57, 58]. A new approximate adder based on the interchangeability of bits is proposed to perform the partial product accumulation. Two approximate error accumulation schemes are used to compensate for the error generated by the approximate adder. The multipliers using these two error reduction schemes are referred to as approximate multiplier 1 (AM1) and approximate multiplier 2 (AM2). The truncation of n LSBs in the partial products in AM1 and AM2 results in truncated approximate multiplier 1 (TAM1) and truncated approximate multiplier 2 (TAM2), respectively.

The Booth algorithm is commonly used for generating partial products in a signed multiplier. The truncated Booth multiplier (TBM), which consists on truncation of LSBs in each operand, has been the most common approach for inducing approximations. Inspired in BAM [51], Farshchi *et al.* omit rows and columns in the array multiplier to reduce the partial product accumulation. This multiplier is denoted in this work as a broken Booth multiplier (BBM) [59].

3.1.2 Automated Design

An automated search-based functional approximation for the design of digital circuits is discussed in [60]. The problem of developing approximate circuits is transformed into a multi-objective design optimization and solved using genetic programming. Mainly, Cartesian genetic programming (CGP) is employed to generate the best trade-off(s) for multiple parameters such as error, area, delay, and power consumption. This methodology is based on an iterative modification of the so-called populations (i.e., a set of candidate solutions), and can be easily integrated into any hardware abstraction

layer (e.g., gate- or register transfer-level) or any semiconductor technology node.

To validate the quality of results of the CGP methodology, an extensive library of approximate multipliers and adders is built in [61]. The *EvoApprox8b* library, contains 473 8-bit adders and 500 8-bit approximate multipliers. The goal of the *EvoApprox8b* library was to have a mean relative error distance (MRED) of less than 10% in the approximate adders and multipliers. Therefore, candidates with the larger errors were discarded. The initial population that was used to generate the approximate adders is based on the following accurate adders: a ripple-carry adder (RCA), a carry-lookahead adder (CLA), a carry-save adder, among other different tree adders. Regarding the approximate multipliers, a ripple-carry array, two variants of a carry-save array, and three variants of an accurate Wallace tree multiplier were employed as the initial population. These accurate designs were coded in Verilog and synthesized using a 180-nm technology.

The main disadvantage of this methodology is the scalability to generate bigger circuits. This is due to the computational cost in order to generate each approximate circuit. However, Mrazek *et al.* demonstrated that the scalability of 8-bit multipliers to build 16-bit multipliers (as shown in Fig. 3.1) is competitive and less computationally expensive than using CGP directly on the 16-bit accurate multipliers [62].

3.1.3 Error Metrics

Different error metrics such as the error rate (ER), the MRED, and the MSE have been employed to quantify the accuracy of the approximate circuits [63, 64, 65, 66]. The ER is defined as the percentage of erroneous outputs among all outputs, the error

distance (ED) is the absolute distance between the approximate and the accurate result, and the mean error distance (MED) is the mean of all possible EDs. The definitions of relative error distance (RED), MRED, and MSE are given below:

$$RED = \frac{|\hat{y} - y|}{y}, \tag{3.1}$$

$$MRED = \frac{\sum_{i=1}^N (RED)}{N}, \tag{3.2}$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2, \tag{3.3}$$

where y , \hat{y} and N denote the accurate result, the approximate result, and the total number of possible input combinations.

3.2 Characterization Under Different Design Constraints

With the growing design complexity of integrated circuits, the same functionality of a circuit needs to run under different operating conditions or scenarios. For instance, a low-power design in wearable devices or smart-phones would be essential to extend the life-time of batteries, whereas a high-performance design would be preferable for video streaming, gaming, or machine learning applications [67]. Therefore, in this section, rather than simply considering the overall power-delay product (PDP), speed and power efficiency are respectively pursued in approximate circuits as independent

design metrics. To achieve this, we take advantage of existing EDA tools to optimize and compare the approximate designs under high-speed and low-power design constraints.

3.2.1 Experimental Setup

Logical representations of 16-bit approximate adders and multipliers are implemented using Verilog and VHDL. Synopsys Design Compiler was employed in the process of synthesis using a 45-nm Nangate process technology [68]. For a fair comparison, all designs use the same design constraint in each library. Throughout the characterization of the high-performance library, the timing constraint was limited to the smallest value of each approximate circuit to optimize delay. The low-power library was synthesized using the lowest value in the section of area design constraint. The “ultra compile” option is used during the synthesis process to maximize the quality of the results. The accuracy of the approximate designs is evaluated with Matlab through Monte Carlo simulations. 10M random inputs with normal distributions were employed to obtain the error metrics described in Section 3.1.3.

3.2.2 Evaluation of Approximate Adders

Simulation Results of High-Performance Circuits

Fig. 3.2 shows the simulation results for the approximate adders synthesized for high-performance. In terms of delay, Fig. 3.2.a shows that, among the adders with small MREDs, Cartesian genetic programming-generated adders (CGPAs) are the fastest. LOA, SCSA, ACAA and specifically GCSA-6 are the fastest for a medium MRED. The highest speed is obtained with CSPA and ESA at the cost of the largest MRED.

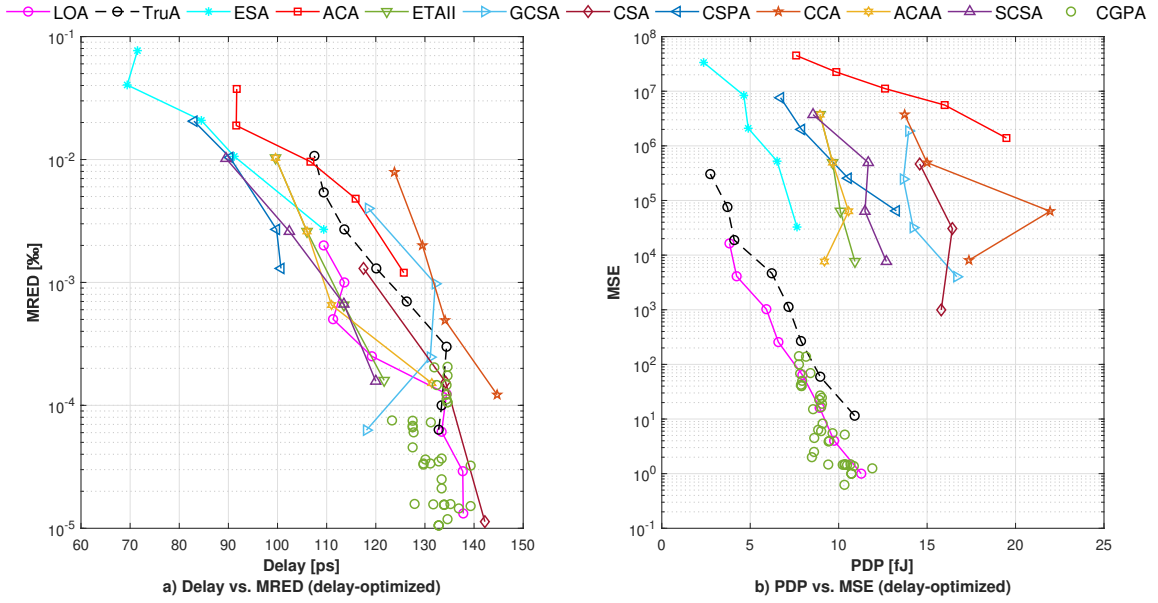


Figure 3.2: A circuit comparison of the 16-bit approximate adders synthesized for high-performance. The parameter k for LOA and TruA ranges from 2 to 9, for ESA and ACA from 8 down to 3 (except 7), for CSA from 5 down to 3, and for the remaining adders from 6 down to 3, all from right to left in the figures. Regarding the CGPAs, the configurations with the lowest error metrics for a specific PDP are reported.

ACA and CCA are the slowest designs for similar MRED compared to other adders. In terms of PDP, Fig. 3.2.b shows that CGPAs are the most efficient designs with the lowest MSE, followed by LOA and TruA. ACA and CCA also remain as the least efficient designs with the largest PDP for a similar MRED compared to other adders.

It is worthwhile to mention that CGPAs were initially implemented for 8 bits. For evaluation purposes in this work, these circuits are scaled to 16 bits where the MSBs are processed using an accurate adder. Although the upper part can also use another CGPA, it may lead to a considerable loss in performance as the 8-bit CGPAs modules do not include a carry-in in the LSBs, which means there would be no carry propagation to the MSBs.

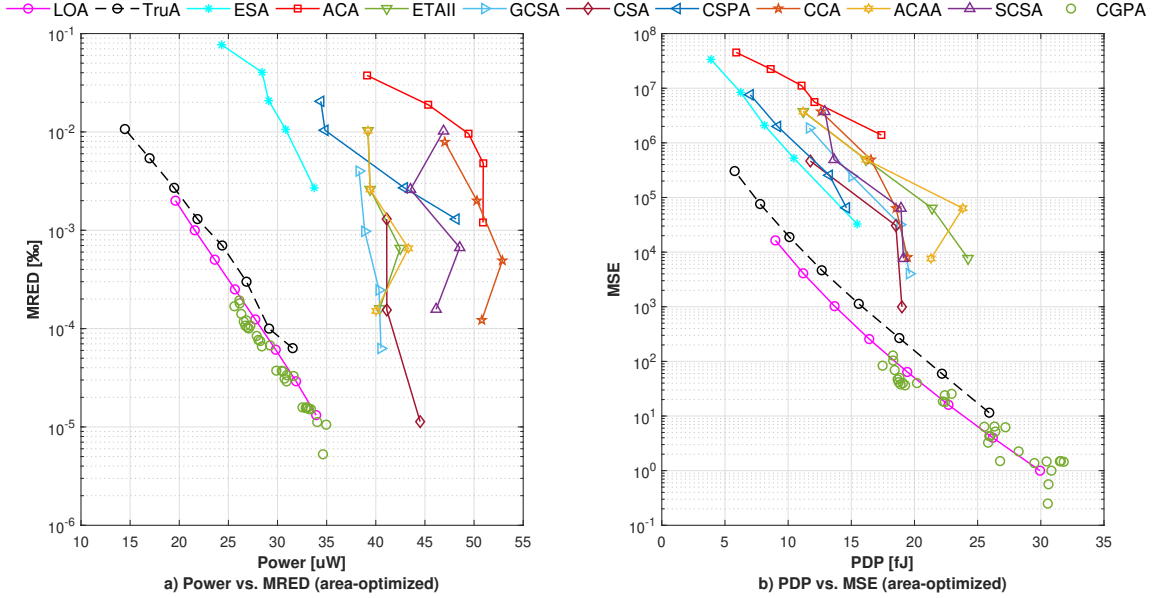


Figure 3.3: A circuit comparison of the 16-bit approximate adders synthesized for low-power. The parameter k for LOA and TruA ranges from 2 to 9, for ESA and ACA from 8 down to 3 (except 7), for CSA from 5 down to 3, and for the remaining adders from 6 down to 3, all from right to left in the figures. Regarding the CGPAs, the configurations with the lowest error metrics for a specific PDP are reported.

Simulation Results of Low-Power Circuits

Fig. 3.3 shows the simulation results for the approximate adders synthesized for low-power. It can be seen that CGPAs, TruA and LOA are the most efficient designs in terms of power. This occurs because these multipliers employ RCAs in the sub-adders, while most of the other approximate designs use CLAs. The ESA consumes a similar power compared to CGPAs, LOA and TruA, but with a higher MRED. On the other hand, CCA, ACAA and ACA are very power hungry. Segmented adders are more competitive against CGPAs, LOA and TruA in PDP rather than only in power. As explained, the reason is that those designs are implemented using CLAs, which run with a higher speed. However, approximations in the LSBs (TruA and LOA) and particularly CGPAs remain as the best design methodologies considering

Table 3.1: Summary of the approximate adders.

Adder	Error Measure		Circuit Measure		
	ER	MSE	Speed	Power	PDP
LOA	HIGH	LOW	LOW	LOW	LOW
TruA	HIGH	-	-	LOW	LOW
ESA	HIGH	HIGH	HIGH	-	-
ACA	-	HIGH	-	HIGH	HIGH
ETAII	LOW	-	-	-	-
GCSA	-	-	-	HIGH	-
CSA	LOW	-	HIGH	-	LOW
CSPA	-	-	-	-	-
CCA	-	-	-	HIGH	-
ACAA	-	HIGH	-	-	-
SCSA	LOW	-	-	-	-
CGPA	HIGH	LOW	LOW	LOW	-

power and speed at the same time.

Finally, a summary of the error and circuit characteristics for the high-performance and low-power approximate adders is shown in Table 3.1.

3.2.3 Evaluation of Approximate Multipliers

Simulation Results of High-Performance Circuits

Fig. 3.4 shows the simulation results for the approximate multipliers optimized for delay. Among all the multipliers, Cartesian genetic programming-generated multipliers (CGPMs) are the most accurate designs. However, the MRED is larger in the CGPMs compared to the other multipliers when it aims for a higher gain in speed. The TruM designs are the fastest unsigned multipliers in the range of low to medium MREDs. TAM1 and AM1 show lower MREDs than TruM at the highest speed due

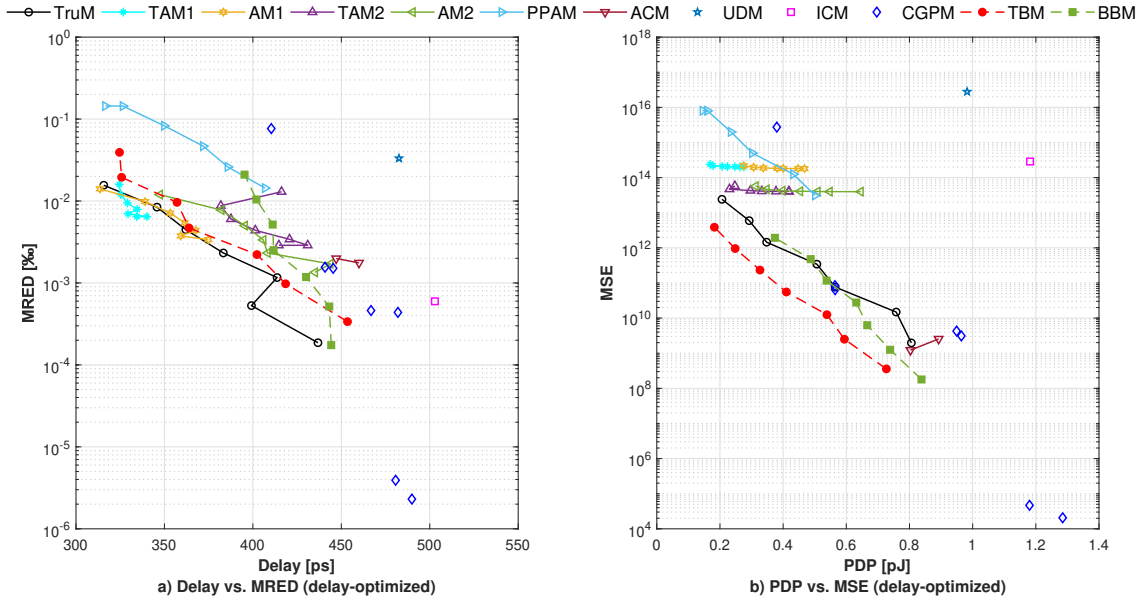


Figure 3.4: A circuit comparison of the 16-bit approximate multipliers synthesized for high-performance. The number of truncated LSBs for TruM, TBM, and BBM varies from 1 to 7 from right to left. The number of MSBs used for error compensation varies from 16 down to 10 for AM1, AM2, TAM1, and TAM2 from right to left. The mode number for ACM is from 4 to 3 from right to left. Regarding PPAM and CGPMs, the best designs in terms of hardware metrics and low-error are reported.

to the introduction of a mechanism for error recovery. Regarding the signed multipliers, the TBM and BBM show similar trends in speed as MRED increases, but TBM outperforms BBM with a smaller error for the same speed.

In terms of PDP (see Fig. 3.4.b), TruM and TBM with the truncation of LSBs and CGPMs are the most efficient multipliers. Although TAM1 and AM1 improved accuracy with a higher speed (see Fig. 3.4.a), they come at the expenses of consuming more power due to the error recovery mechanism. On the contrary, UDM and ICM are the least efficient multipliers with the largest error and smallest gain in circuit performance. For the signed multipliers, TBM shows a better performance than BBM.

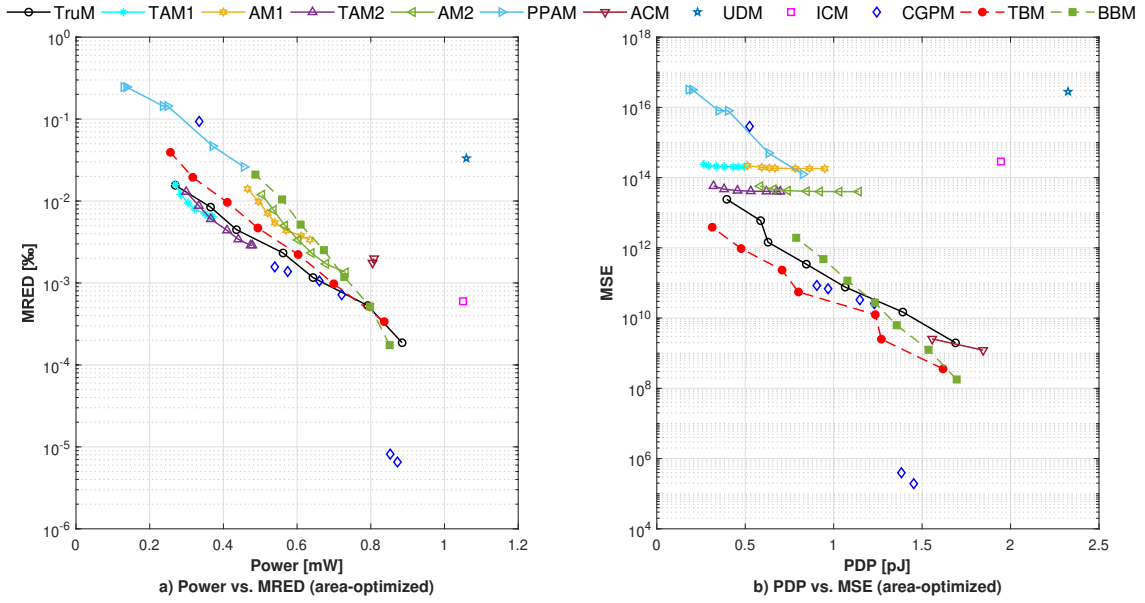


Figure 3.5: A circuit comparison of the 16-bit approximate multipliers synthesized for low-power. The number of truncated LSBs for TruM, TBM, and BBM is from 1 to 7 from right to left. The number of MSBs used for error compensation is from 16 down to 10 for AM1, AM2, TAM1, and TAM2 from right to left. The mode number for ACM is from 4 to 3 from right to left. Regarding PPAM and CGPMs, the best designs in terms of hardware metrics and low-error are reported.

Similar to the CGPAs, CGPMs were originally implemented for 8-bits multipliers. In this experiment, these multipliers were scaled using the methodology proposed in [62]. The results in Fig. 3.4 show that the CGPM designs are competitive (MRED lower than 10^{-5}) when only one 8×8 approximate multiplier is used to construct a 16×16 approximate multiplier (from Fig. 3.1, the M2, M3 and M4 are accurate multipliers, while M1 is the approximate multiplier).

Simulation Results of Low-Power Circuits

Fig. 3.5.a shows that, among all the multipliers with small and medium-range MREDS, CGPMs are the most power-efficient, followed by TruM, AM2 and TAM2.

Table 3.2: Summary of the unsigned approximate multipliers.

Adder	Error Measure		Circuit Measure		
	MRED	MSE	Speed	Power	PDP
TruM	LOW	LOW	-	-	-
TAM1	HIGH	HIGH	HIGH	LOW	LOW
AM1	-	HIGH	-	-	-
TAM2	HIGH	-	HIGH	LOW	LOW
AM2	HIGH	-	-	-	-
PPAM	-	HIGH	-	LOW	-
ACM	-	LOW	LOW	HIGH	-
UDM	-	HIGH	LOW	HIGH	HIGH
ICM	LOW	-	LOW	HIGH	HIGH
CGPM	LOW	LOW	LOW	LOW	LOW

TAM2 and TAM1 are the most accurate designs compared to the other multipliers for low power consumption. UDM and ICM are the least efficient multipliers with the highest error and largest power consumption. Regarding signed multipliers, BBM shows a better performance than TBM in a low value of MRED (10^{-4}). However, TBM starts to outperform BBM in terms of power consumption for a larger MRED. In terms of PDP and MSE (see Fig. 3.5.b), we observed similar trends compared with the high-performance library, except that CGPMs reclaims the highest efficiency for low to medium MRED and MSE. All in all, these CGPMs were originally designed to optimize all the circuit parameters (delay, area and power consumption) together.

Finally, a summary of the error and circuit characteristics for the high-performance and low-power approximate unsigned multipliers is shown in Table 3.2.

3.3 Estimating Error Metrics Under Aging-Induced Delay

As mentioned in Chapter 2, the most advanced process technologies have become more susceptible to delay variability that degrades reliability over time. Among the most critical aspects that affect circuits' reliability, the aging phenomenon is at the forefront. Therefore, in this section, we quantify how the aging-induced delay increases in approximate circuits can result in different errors in the circuit's output due to timing violations. In this scenario, the output of the approximate circuit becomes a function of the current inputs, as well as the previous output value. This study allows us to determine which approximate circuit is most resilient to aging-effects when a remedy to counteract the circuit degradations has not been employed.

3.3.1 Experimental Setup

To obtain the error metrics under aging-induced delay, we run the approximate circuits at their maximum clock frequency determined in the absence of aging. The degradation-aware cell libraries are used to characterize the aging effects in the approximate circuits [69]. We used the Prime Time tool during the process of STA. Gate-level simulations are then executed with ModelSim to obtain and analyze the error metrics. The standard delay file (SDF), which was obtained using Prime Time, is used to induce aging delays in the gate-level simulations.

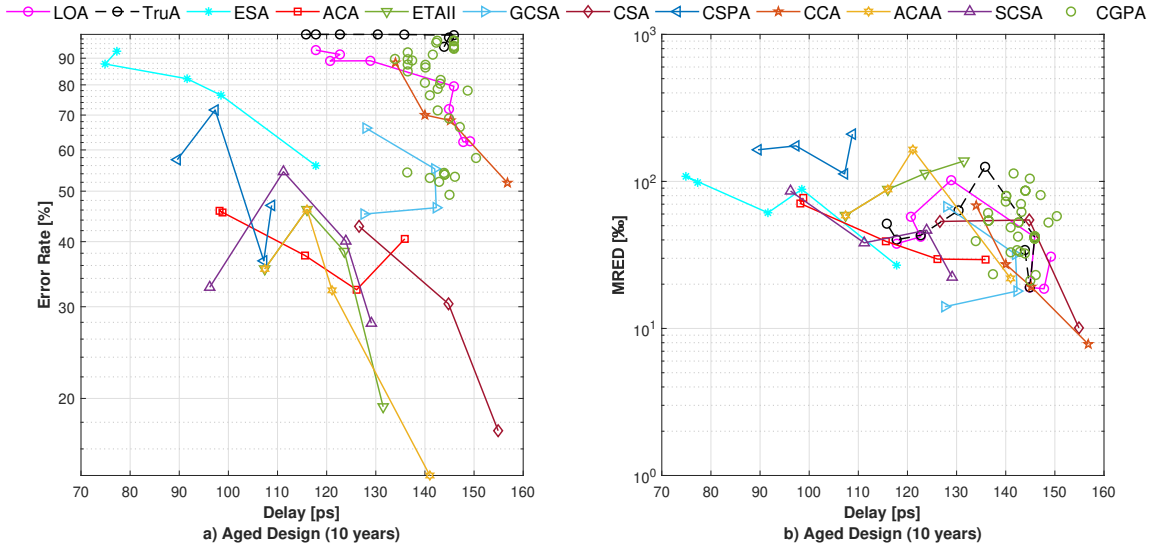


Figure 3.6: Characterization of different errors in the 16-bit approximate adders’ output due to timing violations. The parameter k for LOA and TruA ranges from 2 to 9, for ESA and ACA from 8 down to 3 (except 7), CSA from 5 down to 3 and for the remaining adders from 6 down to 3, all from right to left. Regarding CGPAs, the configurations with the lowest error metrics for a specific delay are reported.

3.3.2 Evaluation of Approximate Adders

Fig. 3.6 shows the results of the error metrics considering aging-induced timing violations. In terms of ER, Fig. 3.6.a shows that ACAA, ETAII and CSA produce more accurate results than the other approximate circuits. For a higher ER, CSPA, ACA and ESA show better performance. Interestingly, design methodologies that introduce errors in the LSBs are less resilient to aging effects, even if the MRED is considered (see Fig. 3.6.b). For instance, LOA, TruA and CGPAs show the highest ER and MRED compared to the other adders. Therefore, segmented adders are more resilient to aging effects compared to designs that use approximations in the LSBs. This is an important finding as one may think that timing violations in multiple blocks (segmented adders) may lead to a higher loss of output quality. However, a larger critical path means a higher level of degradation due to a larger number of

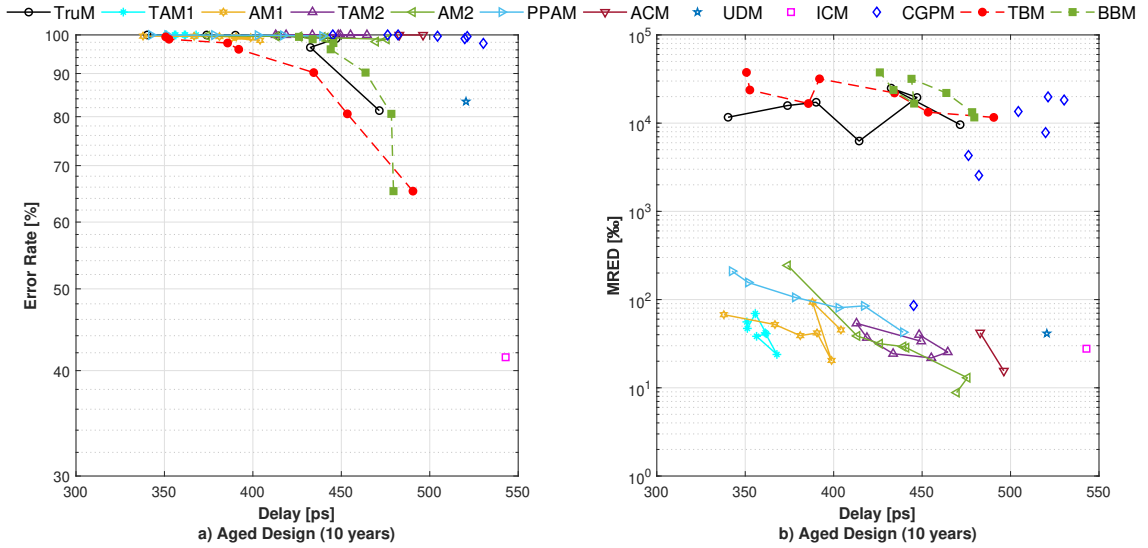


Figure 3.7: Characterization of different errors in the 16-bit approximate multipliers' output due to timing violations. The number of truncated LSBs for TruM, TBM, and BBM varies from 1 to 7 from right to left. The number of MSBs used for error compensation varies from 16 down to 10 for AM1, AM2, TAM1, and TAM2 from right to left. The mode number for ACM varies from 4 to 3 from right to left. Regarding PPAM and CGPMs, the best designs in terms of hardware metrics and low-error are reported.

logic gates, and thus, allowing less time to perform the computations of the MSBs.

3.3.3 Evaluation of Approximate Multipliers

Fig. 3.7 shows the results of the error metrics considering aging-induced timing violations for the approximate multipliers. The ICM multiplier turns out to be the most resilient design towards aging-induced delay with an ER of 41%. Most of the other multipliers reach the 100% of ER, except UDM, TBM and BBM, which are in the range of 65% and 90% (see Fig. 3.7.a). In terms of MRED, approximations in the LSBs result in the worst designs with the largest error. On the other hand, TAM1 and AM1 are the most resilient approximate circuits with the lowest error rates and highest performance. ICM shows a similar MRED as TAM1, but it is less efficient in

terms of speed.

3.4 Characterizing Delay Guard-bands

The results shown in Section 3.3 indicate that delay guard-bands are required to overcome the aging effects in approximate circuits. Otherwise, non-deterministic errors will occur during their lifetime. Therefore, in the following we investigate how the performance of the approximate circuit varies by accurately estimating the required guardband to overcome degradations in each approximate circuit. This is of special interest since recent literature has shown that a critical path in a circuit may not continue to be critical under different levels of workload activity or voltage [70]. Similarly, the work in [4] confirmed that the delay of some cells increases by 70%, while other cells may only be affected by 10% when the temperature rises from a typical value (e.g., $25^{\circ}C$) to the worst-case value ($70^{\circ}C$). In particular, we are interested in an investigation to find out if the performance of the approximate circuits may be different considering the required delay guardband to overcome degradations due to aging and temperature effects.

3.4.1 Simulation Results

Figures 3.8 and 3.9 show the large design flexibility and accuracy range of the approximate adders and multipliers, respectively. Note how the performance (delay) varies according to the levels of stress. Although we found cases where the critical path in a freshly made approximate circuit does not remain as the critical path after transistor degradations, the results indicate that their relative performance in terms

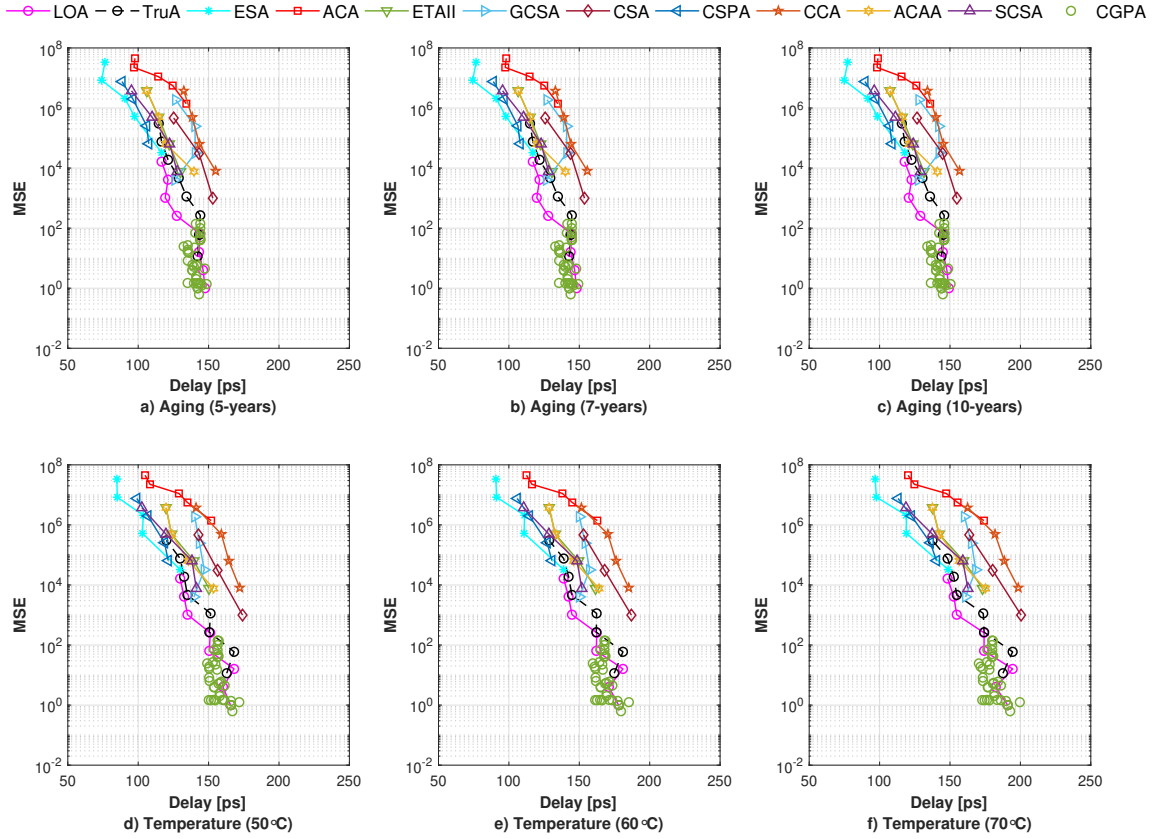


Figure 3.8: Characterizing delay-guardbands of 16-bit high-performance approximate adders under different aging and temperature effects using the proposed methodology in [31]. The parameter k for LOA and TruA ranges from 2 to 9, for ESA and ACA from 8 down to 3 (except 7), CSA from 5 down to 3 and for the remaining adders from 6 down to 3, all from right to left. Regarding the CGPAs, the configurations with the lowest error metrics for a specific delay are reported.

of error remains the same under different levels of degradations. For instance, CGPAs shows the best performance when we aim for an approximate circuit with a low MSE independently of the level of aging or temperature degradation.

The simulation results also indicate that CGPAs is the most efficient circuits for an MSE of up to 10^2 independently of the design constraint. At a larger error, LOA, ESA and CSPA are the most efficient at high-speed and low-power, respectively. With regards to the multipliers, CGPMs shows the best performance at a given

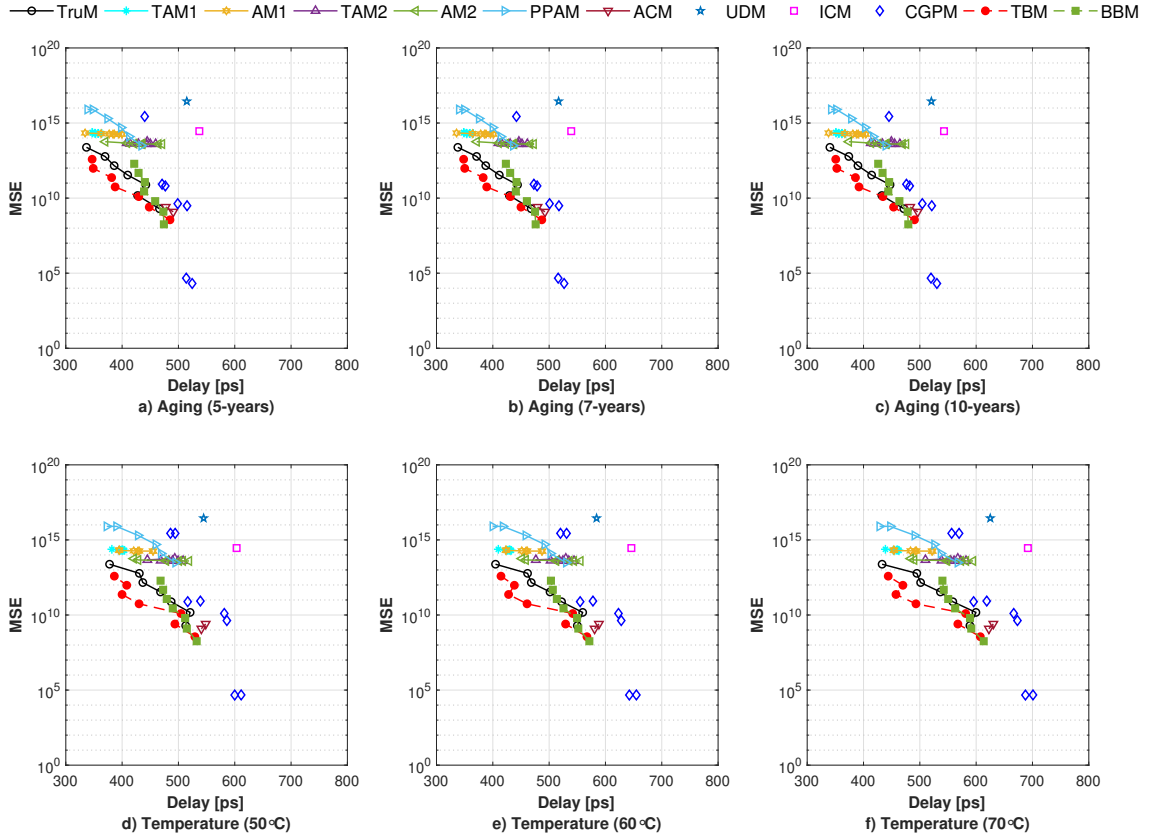


Figure 3.9: Characterizing delay-guardbands of 16-bit high-performance approximate multipliers under different aging and temperature effects using the proposed methodology in [31]. The number of truncated LSBs for TruM, TBM, and BBM is from 1 to 7 from right to left. The number of MSBs used for error compensation is from 16 down to 10 for AM1, AM2, TAM1, and TAM2 from right to left. The mode number for ACM is from 4 to 3 from right to left. Regarding PPAM and CGPMs, the best designs are reported.

low MSE, followed by TruM. But, TAM2 and TAM1 become more effective for a larger MSE. Similar results for both approximate adders and multipliers were observed considering the other error metrics such as the MRED and normalized mean error distance (NMED).

3.5 Summary

As a first study of this kind, this chapter presents an exhaustive evaluation of approximate arithmetic circuits under different workload scenarios or circuit requirements. First, we assessed the performance of approximate circuits (without any degradations) under different design constraints, i.e., high-speed and low-power. Next, we evaluated the impact on the performance of the approximate circuits considering aging-induced timing violations (in 10 years). Finally, we assessed the performance of approximate adders and multipliers, employing a state-of-the-art guardband technique. This allowed us to precisely evaluate actual trade-offs in approximate circuits considering the required guard-band to sustain lifetime reliability.

The simulation results confirm that automatically generated approximate design methodology has made substantial progress in obtaining significant gains in power without having to incur a large error. However, performance continues as one of the main drawbacks of this methodology. Despite the fact that CGPAs outperform the other approximate adders, independently of the design constraints, the CGPMs are not as effective as the manual designs for high-speed operation. UDM and ICM continue to be the least efficient multipliers for either low-power or high-performance designs. The scheme of truncation of LSBs (e.g., TruA, TruM, and TBM) remains as a suitable option to enhance performance and energy efficiency in a controllable and straightforward manner to introducing approximations.

Chapter 4

Trading-off degradations for approximations at the Component Level

In this chapter, the design methodology proposed in [38] is improved and employed in a larger set of approximate circuits to determine optimal solutions towards degradation-induced approximation. This methodology, rather than using delay guard-bands to guarantee reliability, consists of converting degradations to deterministic and controllable errors coming solely from an approximate arithmetic circuit. Specifically, an approximate circuit is characterized by considering possible degradations, to obtain the same effective performance as an accurate circuit without considering any degradation. This can be understood as maintaining the performance of an accurate arithmetic component at the cost of a quality loss in the output. By this means, no guardband is required at the component level to maintain reliability during its lifetime. Section 4.1 explains this methodology in more detail. Sections 4.2 and 4.3

employ this methodology to overcome aging and temperature effects, respectively.

4.1 Design Methodology

The methodology to trade-off degradations for precision is summarized in Fig. 4.1. Different from [38], we present a comprehensive framework using the main characteristics of approximate computing. The most important steps are summarized below.

1) Logic Synthesis:

It is known that an RCA is usually used for area minimization and the CLA is used for delay minimization. While a CLA outperforms RCA in terms of delay, this comes at the expense of higher energy consumption. Therefore, to achieve the optimal performance of each approximate circuit under different design constraints, circuits are coded in the HDL with a high-level description (identified using “+”) when it is possible. For instance, the final addition for most of the approximate multipliers at the partial product accumulation stage uses the “+” operation. By employing this coding strategy during the design phase and the “compile_ultra” option in the Design Compiler tool, we can obtain a higher quality of results using aggressive and efficient optimization algorithms from the Synopsys tools instead of exploring approximate circuits with different adder tree topologies [28].

2) Verifying Timing Across Degradations:

An exhaustive timing verification of all possible scenarios is impractical, and so the use of Static Timing Analysis (STA) with Prime Time tool was used to maximize the performance and efficiently verify the required timing margin to overcome different

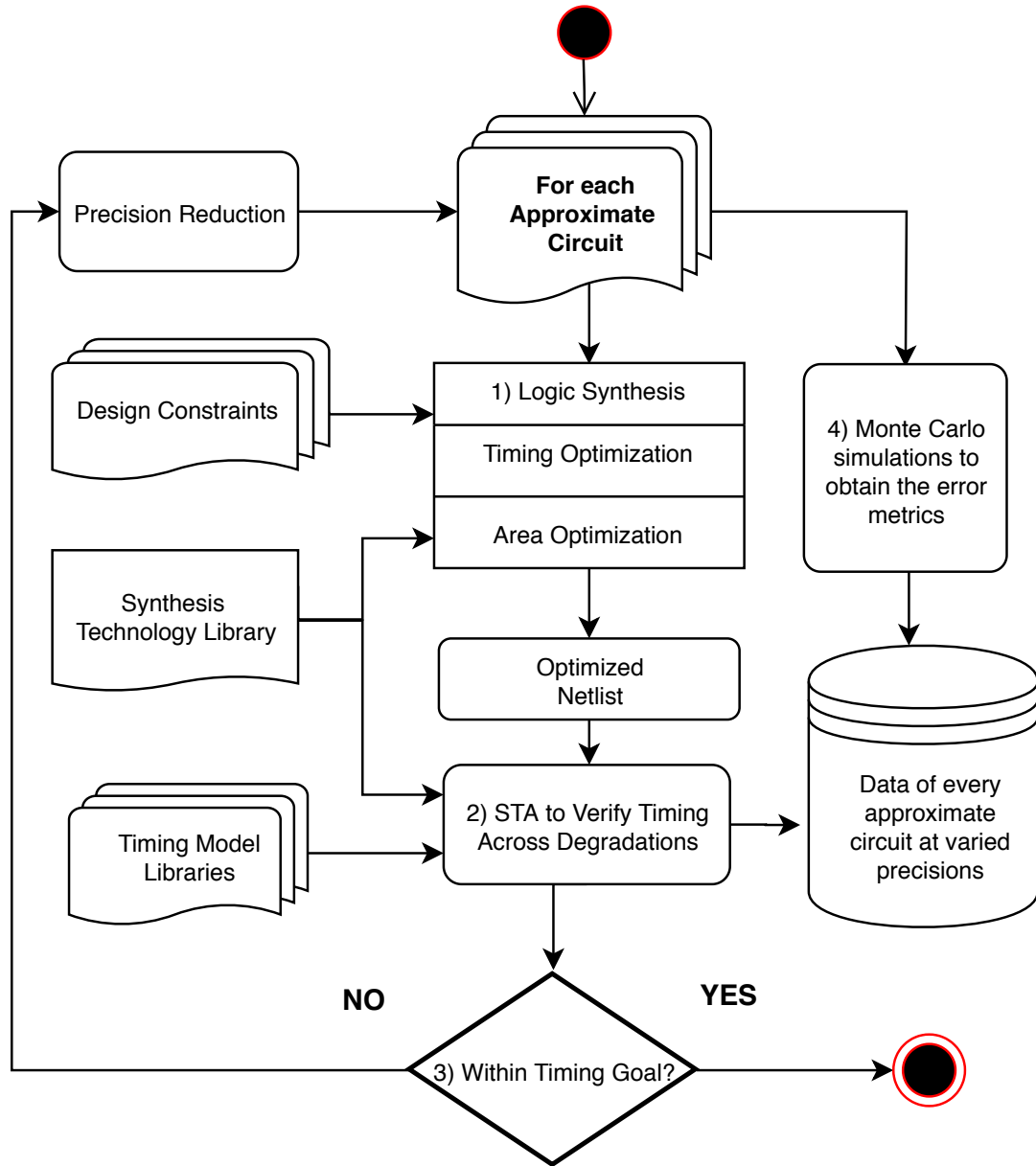


Figure 4.1: Design methodology at the component level to convert degradations to controllable errors using approximate circuits.

degradations in the optimized netlist [71]. Timing model libraries are required to provide detailed information about the cells under different stress conditions. While some technology libraries include timing models of Process-Voltage-Temperature (PVT)

variations, other timing models that describe how transistors degrade considering aging or temperature are publicly available in [69]. The output of this stage will be an estimation of the required time to avoid violations in the critical path when the chip performance is degraded.

3) Timing Goal:

The objective is to accurately trade off degradations of an accurate circuit for a quality loss, rather than using delay guard-bands to overcome the degradations. Therefore, if the delay of the approximate circuit under respective degradations is larger than the delay of the timing goal (set for the accurate circuit without degradations), then the whole process is repeated by reducing the precision in the approximate circuit. We assumed that gains in performance are obtained each time we reduce the precision in the approximate circuits.

4) Obtaining Error Metrics:

The output quality of an approximate circuit is typically expressed using one or several error metrics (see Section 3.1.3). The selection of the right metrics is a key step during the process of evaluation at the component level. For instance, an arithmetic error metric (e.g., MRED or MSE) would often be more useful than the ER to evaluate the impact on a target application. On the other hand, obtaining several error metrics with all possible input combinations may be overly time-consuming and computationally expensive. As a practical strategy, Monte Carlo simulations can be employed to evaluate the functionality of each approximate circuit design. This statistical technique applies a randomly selected subset of the set of all possible input

vectors based on certain probability distributions (e.g., uniform, Poisson, Gaussian, or exponential). In the context of this thesis, we employed 10 million uniformly distributed random input combinations to evaluate the 16-bit approximate multipliers and adders. The error metrics obtained with the Monte Carlo simulations could be stored in a database (in addition to the circuit's metrics) for further comparison at the architectural level (see Section 5.3).

4.2 Towards Aging-Induced Approximation

4.2.1 Experimental Setup

The 45-nm NanGate process technology with a supply voltage of $1.2V$ is employed during the process of synthesis [68]. In this experiment, we characterize approximation circuits towards aging-induced approximation using two different design constraints: high-performance and low-power. The degradation-aware cell libraries are employed during the STA with Prime Time to accurately estimate the effects of aging in the circuits [69]. To be competitive with state-of-the-art methods, the timing goal is defined as the clock frequency given by the accurate circuit from the Synopsys DesignWare library in the absence of aging. Finally, the MSE, which is correlated with the PSNR, is used for ranking the approximate circuits and determining the best design.

4.2.2 Simulation Results

As mentioned, the objective is to maintain the performance of an accurate arithmetic component during its lifetime at the cost of a quality loss, rather than using delay

Table 4.1: Characterizing 16-bit high-performance approximate circuits towards aging-induced approximation. The circuits are ranked by the MSE from the lowest to the highest value.

Component	Precision	Delay (<i>ps</i>)		ER (%)	MRED (10^3)	MSE
		0 year	10 years			
Adder	Accurate	138.60	149.50	0	0.0	0.00E+00
	CGPA-156	127.90	136.40	43.74	0.16	1.49E+00
	LOA-6	126.30	135.80	82.21	0.25	2.56E+02
	TruA-5	126.30	135.80	99.90	0.70	1.13E+03
	ETAII-6	121.70	131.50	0.73	0.16	7.63E+03
	SCSA-6	120.00	129.10	0.72	0.16	7.64E+03
	ESA-8	109.40	117.80	49.83	2.70	3.26E+04
	ACAA-5	111.00	121.10	2.29	0.65	6.34E+04
	CSPA-6	100.70	108.80	9.13	1.30	6.46E+04
	CSA-3	117.50	126.60	1.76	1.30	4.64E+05
	ACA-8	125.70	135.90	0.68	1.20	1.39E+06
	GCSA-3	118.50	128.00	10.76	4.00	1.86E+06
	CCA-3	123.80	134.00	23.66	7.90	3.73E+06
Unsigned multipliers	Accurate	457.70	489.60	0.00	0.00	0.00E+00
	TruM-2	399.20	432.50	93.75	0.53	1.48E+10
	PPAM-JOK8	406.90	439.30	99.61	14.37	3.10E+13
	AM2-14	408.30	441.00	99.10	2.32	4.02E+13
	TAM2-16	414.70	448.30	99.99	2.88	4.02E+13
	AM1-16	374.80	404.10	98.22	3.37	1.80E+14
	TAM1-16	334.40	361.30	99.99	6.45	2.02E+14
CGPM-4A19	399.00	429.50	99.99	82.16	1.89E+15	
Signed multipliers	Accurate	462.90	500.20	0.00	0.00	0.00E+00
	TBM-2	418.50	453.40	93.74	0.98	2.50E+09
	BBM-4	411.70	444.00	93.74	2.51	2.77E+10

guard-bands. Tables 4.1 and 4.2 show the required level of precision of each approximate circuit to achieve this goal. Note that the worst-case aging-induced delay (in 10 years) for the approximate circuits is lower than the delay of the accurate circuit

in the absence of aging (0 year). Therefore, we can use any of these approximate circuits instead of the accurate one with the assurance that delay guardbands would not be required to ensure circuit accuracy after 10 years.

As can be seen for the adders synthesized for high-performance (see Table 4.1), CGPA-156 is the best design towards aging-induced approximation. The CGPAs do not only show the lowest MSE, but also the lowest MRED among all the approximate adders. Considering only the manual designs, LOA-6 and TruA-5 are the most effective circuits. Although LOA outperforms TruA in terms of error metrics, it comes at the cost of an insignificant increase in area and power. On the other hand, if ER is the most important measure for a target application, ETAII and GCSA (with an ER less than 1%) considerably outperform CGPA-156, LOA-6 and TruA-5. The use of these speculative adders (ETAII-6 and GCSA-3) is highly recommended in applications where the probability of a large carry chain is relatively small [49].

Regarding the high-performance multipliers, a TruM shows the best performance with the lowest values in the MSE, MRED and ER. The results indicate that the truncation of 2 bits is enough to ensure the target reliability for 10 years. The circuit CGPM-4A19¹ is not as efficient as the manual designs toward aging-induced approximation. The CGPMs were originally designed to improve power and delay (PDP) simultaneously, which limits the gains in speed. In terms of signed multipliers, experimental results show that TBM-2, with lower error values, is more effective than BBM.

Table 4.2 shows the results towards-aging induced approximation for the circuits synthesized for low-power. Different from the high-performance circuits, we observed that automatically generated designs outperform manual designs. Regarding the

¹4A19 represents the architecture A4 with the circuit 19 following the nomenclature in [62].

approximate adders, the CGPA-449 has the lowest MRED and MSE values, followed by LOA-3 and TruA-2. The less-effective designs in terms of MSE are the CSPA-6 and ACA-8. Regarding the approximate unsigned multipliers, we observed that CGPM-1A222² outperforms TruA-2 with a substantial difference in the MSE. AM1-16 and TAM1-16 are the less effective unsigned multipliers in terms of error; however, the gains in speed are much larger than the other circuits.

4.3 Towards Temperature-Induced Approximation

High temperatures boost the aging phenomena and, in turn, make the circuits less resilient. Moreover, the so-called dark silicon problem [24] limits the full performance to that of only a sub-set of processing cores within an on-chip system to meet the TDP constraint. The worst-case temperature is typically determined by the capability of the cooling technology. Expensive cooling systems have led to the employment of larger guardbands to compensate the noticeable slowness in transistors [4]. However, this may not be always appropriate as modern applications, such as machine learning, require high-performance chips. Therefore, in this section we also compensate for the worst-case temperature scenario with a controllable quality loss in arithmetic components instead without compromising a loss in performance.

4.3.1 Experimental Setup

During the characterization of temperature-induced approximation, we synthesized the accurate circuit from the Synopsys DesignWare library under a nominal temperature of 25°C and a supply voltage of 1.2V. The maximum clock frequency is used

²1A22 represents the architecture A1 with the circuit 222 following the nomenclature in [62].

Table 4.2: Characterizing 16-bit low-power approximate circuits towards aging-induced approximation. The circuits are ranked by the MSE from the lowest to the highest value.

Component	Precision	Delay (ps)	Delay (ps)	ER	MRED	MSE
		0 year	10 years	(%)	(10^3)	
Adder	Accurate	942.90	1023.30	0.00	0.00	0.00E+00
	CGPA-449	821.70	916.90	34.37	0.01	2.25E+00
	LOA-3	821.70	891.80	57.80	0.03	4.00E+00
	TruA-2	821.70	891.80	93.74	0.06	1.15E+01
	CSA-5	427.00	457.40	0.02	0.01	9.93E+02
	GCSA-6	483.70	523.20	0.74	0.06	4.02E+03
	ETAIL-6	600.00	643.30	0.73	0.16	7.63E+03
	SCSA-6	414.60	447.60	0.72	0.16	7.64E+03
	ACAA-6	532.40	577.30	0.72	0.15	7.64E+03
	CCA-6	382.80	413.00	1.50	0.12	8.05E+03
	ESA-8	458.40	497.10	49.83	2.70	3.26E+04
	CSPA-6	303.50	326.40	9.13	1.30	6.46E+04
ACA-8	341.40	365.80	0.68	1.20	1.39E+06	
Unsigned multiplier	Accurate	2009.30	2177.00	0.00	0.00	0.00E+00
	CGPM-1A222	1665.40	1802.10	0.03	0.01	1.92E+05
	TrunM-2	1757.00	1903.30	93.75	0.53	1.48E+10
	AM2-16	1560.10	1693.00	97.95	1.35	3.97E+13
	TAM2-16	1459.20	1580.20	99.99	2.88	4.02E+13
	PPAM-JOK9	1811.70	1982.10	99.80	26.08	1.24E+14
	AM1-16	1496.10	1622.60	98.22	3.37	1.80E+14
	TAM1-16	1336.90	1441.00	99.99	6.45	2.02E+14
Signed multipliers	Accurate	2059.70	2234.30	0.00	0.00	0.00E+00
	TBM-2	1815.20	1969.00	93.74	0.98	2.50E+09
	BBM-3	1863.50	2022.10	87.48	1.18	6.26E+09

to define the timing goal towards temperature-induced approximation. We also synthesized all the approximate circuits under the same conditions. High-performance and low-power libraries were investigated using different design constraints. In the

Table 4.3: Characterizing 16-bit high-performance approximate circuits towards temperature-induced approximation. The circuits are ranked by the MSE from the lowest to the highest value.

Component	Precision	Delay (<i>ps</i>)	Delay (<i>ps</i>)	ER	MRED	MSE
		25 °C	70 °C	(%)	(10 ³)	
Adder	Accurate	143.50	196.70	0.00	0.00	0.00E+00
	LOA-10	100.10	137.60	94.36	4.00	6.55E+04
	CSPA-5	99.20	136.00	11.31	2.70	2.55E+05
	TruA-9	100.10	137.60	99.99	10.70	3.04E+05
	ESA-6	86.30	119.10	73.04	10.60	5.23E+05
	ACAA-3	99.70	137.60	18.86	10.30	3.73E+06
	ETAII-3	99.70	137.60	18.91	10.30	3.73E+06
	SCSA-3	85.40	118.70	18.89	10.20	3.73E+06
	ACA-4	90.40	124.70	16.65	18.90	2.24E+07
Unsigned multipliers	Accurate	467.40	643.80	0.00	0.00	0.00E+00
	TruM-7	315.90	433.00	99.99	15.57	2.40E+13
	AM1-11	330.50	453.90	99.59	9.85	1.97E+14
	TAM1-16	335.10	461.70	99.98	6.45	2.02E+14
	PPAM-J1k11	326.50	351.70	99.95	144.20	8.00E+15
Signed multipliers	Accurate	460.00	632.00	0.00	0.00	0.00E+00
	TBM-7	322.70	443.90	99.99	39.23	3.86E+12

process, we used the temperature-aware cell libraries from [4]. A thermal-aware timing analysis was performed with Prime Time to accurately estimate the temperature degradations in the approximate circuits.

4.3.2 Simulation Results

Tables 4.3 and 4.4 show the required level of precision of each approximate circuit towards temperature-induced approximation. Note that the worst-case temperature delay (at 70 °C) for the approximate circuits is lower than the delay of the accurate

design in a nominal temperature (25 °C). We also used the criteria of the lowest MSE in the temperature-induced delay approximate libraries to rank the approximate circuits in the corresponding tables. Different from the aging-induced approximation, we observed that some approximate design methodologies do not meet the timing goal when the circuit is degraded at temperatures higher than 60°C, which means that those circuits still require a delay guard-band on the top of the maximum clock frequency (determined by the accurate circuit) to guarantee reliability.

Table 4.3 shows the results for the high-performance circuits. As can be seen, automatically generated designs (CGPAs and CGPMs) are not present in the results. Regarding the manual designs, LOA-10 shows the lowest MSE towards temperature-induced approximation at 70°C. CSPA-5 shows a larger MSE than LOA-10, but the MRED and ER are lower. In terms of approximate unsigned multipliers, the truncation of 7 bits (TruM-7) results in the best design followed by AM1-11. Although TAM1-16 shows a relatively large MSE, this approximate design has the lowest MRED. Regarding the signed multipliers, the truncation of 7 bits in the booth Multiplier (TBM-7) provides the best trade-off for degradation and performance compared to the BBM (not present in the table).

Table 4.4 shows the results for the circuits synthesized for low-power. Similar to the high performance circuits, the automated designs are absent. Regarding the manual designs, LOA-6 shows the best performance with the lowest MSE. However, CSPA-5 has lower MRED and ER. In terms of approximate unsigned multipliers, truncation of 7 bits (TruM-7) is the best design followed by AM1-11. Although TAM1-16 shows a relatively large MSE, this approximate design has the lowest MRED. Regarding the signed multipliers, the TBM-7 provides the best trade-off towards

4.3. TOWARDS TEMPERATURE-INDUCED APPROXIMATION 54

Table 4.4: Characterizing 16-bit low-power approximate circuits towards temperature induced approximation. The circuits are ranked by the MSE from the lowest to the highest value.

Component	Precision	Delay (<i>ps</i>)		ER (%)	MRED (10^3)	MSE
		25 °C	70 °C			
Adder	Accurate	942.80	1303.40	0.00	0.00	0.00E+00
	LOA-6	639.70	885.00	82.22	0.25	2.56E+02
	CSA-5	426.10	580.50	0.62	0.01	9.93E+02
	TruA-5	639.70	885.00	99.90	0.70	1.13E+03
	GCSA-6	483.70	664.30	0.74	0.06	4.02E+03
	ETAIL-6	597.70	815.00	0.73	0.16	7.63E+03
	SCSA-6	316.80	436.10	0.72	0.16	7.64E+03
	ACAA-6	532.10	734.60	0.72	0.15	7.64E+03
	CCA-6	338.10	466.60	1.49	0.12	8.05E+03
	ESA-8	458.60	633.90	49.83	2.70	3.26E+04
	CSPA-6	303.70	416.60	9.13	1.30	6.46E+04
	ACA-8	340.80	463.30	0.68	1.20	1.39E+06
Unsigned multipliers	Accurate	2009.10	2784.70	0.00	0.00	0.00E+00
	TruM-5	1443.5	1997.00	99.89	4.47	1.44E+12
	AM2-14	1444.40	1998.40	99.10	2.32	4.02E+13
	TAM2-16	1459.20	2003.50	99.98	2.88	4.02E+13
	AM1-15	1412.10	1949.30	98.81	3.75	1.80E+14
	TAM1-16	1330.90	1818.90	99.99	6.45	2.02E+14
	PPAM-J0K13	1390.60	1917.70	99.99	245.96	3.20E+16
Signed multipliers	Accurate	2053.10	2827.30	0.00	0.00	0.00E+00
	TBM-7	1223.60	1684.40	99.99	39.23	3.86E+12

temperature-induced approximation.

4.4 Summary

The aging phenomena affect transistor speed over time. On the other hand, a high-speed is frequently tied to the TDP constraint. Both have led to including delay-guardband within the clock period to improve the reliability at the cost of a significant loss in circuit performance. Differently from this conventional technique, we described a methodology to remove guardbands while maintaining hardware performance. Delay guardbands are converted into deterministic and controlled approximations coming solely from arithmetic units. This methodology employs an state-of-the-art technique that accurately quantify the impact of transistor degradations in combination with the use of aggressive synthesis algorithms from the Synopsys tools. This allows us to accurately trade-off optimum guardband selection for the minimum loss in precision in the approximate circuits.

To claim the optimum solution, a large number of different approximate arithmetic circuits were evaluated. We demonstrated that different levels of approximation are required to overcome degradations under different workload scenarios (aging or temperature) or circuit requirements (high-performance or low-power). Interestingly, we found that the truncation of LSBs is not always the most effective technique towards degradation-induced approximation. This is an important finding since current research work has been exclusively using this technique to trade-off degradations for a quality loss in the approximate arithmetic circuits.

We concluded for the high-performance approximate adders that CGPAs and LOA have the lowest error metrics among all the approximate circuits when we aimed to mitigate small degradations (aging-induced timing errors). Most of the approximate adders are designed for a high-speed operation (e.g., CSPA-5 and CSA-5), which made

them suitable to overcome larger degradations (e.g. temperatures beyond $70^{\circ}C$). However, these designs generate higher power dissipation than approximations in the LSBs. The simulation results show similar trends for the approximate adders synthesized for low-power operation. Regarding the approximate multipliers, the truncation of LSBs has the lowest MSE towards degradation-induced approximation, independently of the workload scenario or circuit requirement. However, if the MRED is considered as the most important error metric instead of MSE, AM2, AM1 and TAM2 are more effective approximate designs.

Chapter 5

Trading-off degradations for approximations at the Architectural Level

Up to this point, the impact of degradations on circuits has been exclusively investigated at the component level. However, heterogeneous designs now incorporate multiple functional components, which results in a multi-billion gate chips containing millions of connections. This has increased the difficulty of integrating reliability methodologies at the system level [72]. However, we demonstrate in this chapter how degradation-induced delay increases at the component level can be converted into approximations at the architectural level in an effective way. To validate the proposed methodology, three different image processing applications are evaluated independently.

5.1 Image Processing Applications

The need to support image processing applications on energy- and speed-constrained devices has steadily grown. These applications considerably perform arithmetic operations such as addition and multiplication, which usually limits the hardware performance. Recently, hardware accelerators (see Fig. 1.1 in Chapter 1) have emerged as a solution to improve the performance of image processing algorithms with a large number of parallel units. However, the performance of hardware accelerators is also limited for the TDP constraint as we demonstrated in Section 5.2. In this chapter, hardware architectures for an inverse discrete cosine transform (IDCT), an image sharpening and an image smoothing application are implemented in register transfer level (RTL) to improve the performance considering the effects of temperature. In the end, we demonstrated how to accurately exploit the degree of error tolerance for these applications, while still aiming to guarantee timing correctness throughout the chip lifetime without a performance loss.

5.1.1 DCT/IDCT

The discrete cosine transform (DCT) is widely used for image compression. In lossy compression, the image quality is compromised to be stored or transmitted in an efficient way. The DCT algorithm is given by [73]:

$$\mathbf{F}[u, v] = \frac{1}{64} \sum_{m=0}^7 \sum_{n=0}^7 \mathbf{f}[m, n] \cos \left[\frac{(2m+1)u\pi}{16} \right] \cos \left[\frac{(2n+1)v\pi}{16} \right], \quad (5.1)$$

where $\mathbf{f}[m, n]$ indicates the 8×8 pixel blocks of the image input, and $\mathbf{F}[u, v]$ is the 8×8 DCT output.

Finally, the image can be reconstructed using the inverse DCT as follows:

$$\mathbf{f}[\mathbf{m}, \mathbf{n}] = \sum_{u=0}^7 \sum_{v=0}^7 c[u] c[v] \mathbf{F}[u, v] \cos \left[\frac{(2m+1)u\pi}{16} \right] \cos \left[\frac{(2n+1)v\pi}{16} \right], \quad (5.2)$$

where $c[\lambda] = 1$ for $\lambda = 0$, and $c[\lambda] = 2$ for $\lambda = 1, 2, 3, \dots, 7$.

5.1.2 Image Smoothing

An image smoothing algorithm is commonly used to reduce the noise within an image.

The smoothed image \mathbf{Y} is computed by [74]:

$$\mathbf{Y}(x, y) = \frac{1}{960} \sum_{i=-2}^2 \sum_{j=-2}^2 \mathbf{G}(i+3, i+3) \mathbf{I}(x-i, y-j), \quad (5.3)$$

where \mathbf{I} represents the input image, and \mathbf{G} is a 5×5 matrix given by

$$\mathbf{G} = \begin{bmatrix} 16 & 16 & 16 & 16 & 16 \\ 16 & 64 & 64 & 64 & 16 \\ 16 & 64 & 192 & 64 & 16 \\ 16 & 64 & 64 & 64 & 16 \\ 16 & 16 & 16 & 16 & 16 \end{bmatrix}. \quad (5.4)$$

5.1.3 Image Sharpening

An image sharpening algorithm is widely employed in image processing applications to sharpen blurred images. The sharpened image \mathbf{S} is computed by $\mathbf{S} = 2 \mathbf{I}(x,y) - \mathbf{Y}(x,y)$, where $\mathbf{I}(x,y)$ denotes a pixel in the original image, \mathbf{S} is the resulting sharpened

image, and \mathbf{Y} is given by:

$$\mathbf{Y}(x, y) = \frac{1}{4368} \sum_{i=-2}^2 \sum_{j=-2}^2 \mathbf{G}(i+3, i+3) \mathbf{I}(x-i, y-j), \quad (5.5)$$

where \mathbf{G} is a 5×5 matrix given by

$$G = \begin{bmatrix} 16 & 64 & 112 & 64 & 16 \\ 64 & 256 & 416 & 256 & 64 \\ 112 & 416 & 656 & 416 & 112 \\ 64 & 256 & 416 & 256 & 64 \\ 16 & 64 & 112 & 64 & 16 \end{bmatrix}. \quad (5.6)$$

Observing that multiplication of \mathbf{I} by 2 can be performed by bit-shifting, this arithmetic operation is not required to obtain \mathbf{S} from \mathbf{Y} [75].

5.2 Motivational Study Case

First, we evaluate the image processing applications towards temperature-induced delay. Fig. 5.1 shows the ideal outputs of an IDCT, image smoothing and image sharpening applications when the chip is working at the nominal temperature ($25^\circ C$). However, Fig. 5.2 shows how the output considerably degrades when the circuits are exposed to a temperature of $70^\circ C$ and no remedy is employed in the circuits to counteract the slow-down of the transistors. By this means, we show that guard-bands are required to ensure reliability even for error-tolerant applications. As we discussed

in Section 2.2, most of the current guard-band techniques incur a significant performance loss to deal with these catastrophic events. In the following section, we present a methodology for trading degradations for approximations at the architectural level. This methodology, rather than using delay guard-bands to sustain reliability, employs approximations to compensate the temperature effects.



Figure 5.1: Image quality output of three different image processing applications when the chip is working in nominal conditions (25°C).

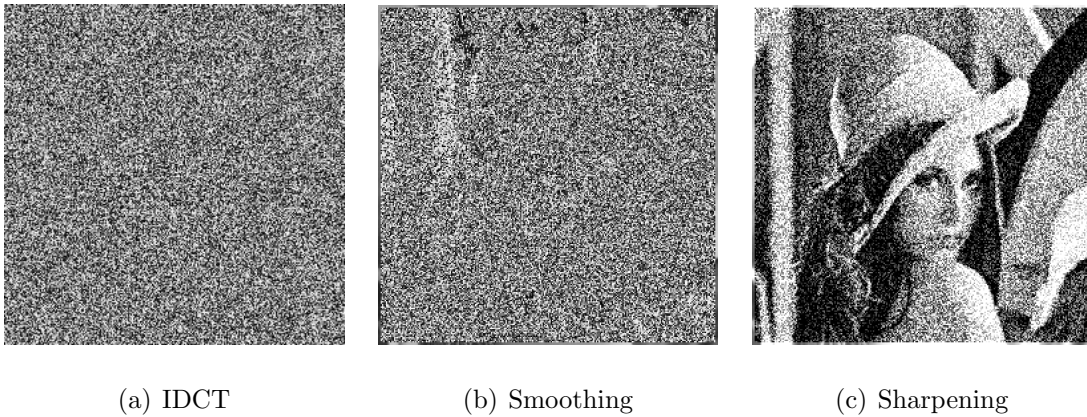


Figure 5.2: Image quality output of three different image processing applications after the circuit is exposed at 70°C without a technique to overcome the transistor degradations.

5.3 Design Methodology

The process for converting degradations into controllable errors for an architectural-level design can be divided into multiple segments, as shown in Fig. 5.3. In this chapter, we exclusively applied the design methodology to overcome aging and temperature degradations in the arithmetic computation. However, this methodology can also be applied to reduce the significant increase of delay in the die due to the place-and-route stage or On-Chip Variations (OCV) [71]. This methodology has been taken and improved from [38] to optimize the final result by means of design exploration. In the following, we explain in detail the methodology step by step.

1) Obtaining Timing Constraints

First, the architecture was synthesized to obtain the critical path (CP) delay in the absence of any degradation ($t_{CP}(freshDesign)$). This delay represents the required timing constraint that the whole design must fulfill under the targeted aging or temperature stress condition. At this point, it is assumed that the critical path belongs to the arithmetic circuits with the purpose of introducing approximations in the computations, and thus, reduce the critical path delay. Other components, such as control units, can be protected through traditional techniques, such as using stronger gates [29].

2) Estimating Degradations

Under a specific level of stress, STA is performed to the whole design to obtain the delay of every combinational datapath block (B_k) within the netlist ($t_{B_k}(postStress)$). This allows us to calculate the available timing slack $t_{B_k}(slack)$ (see Equation 5.7)

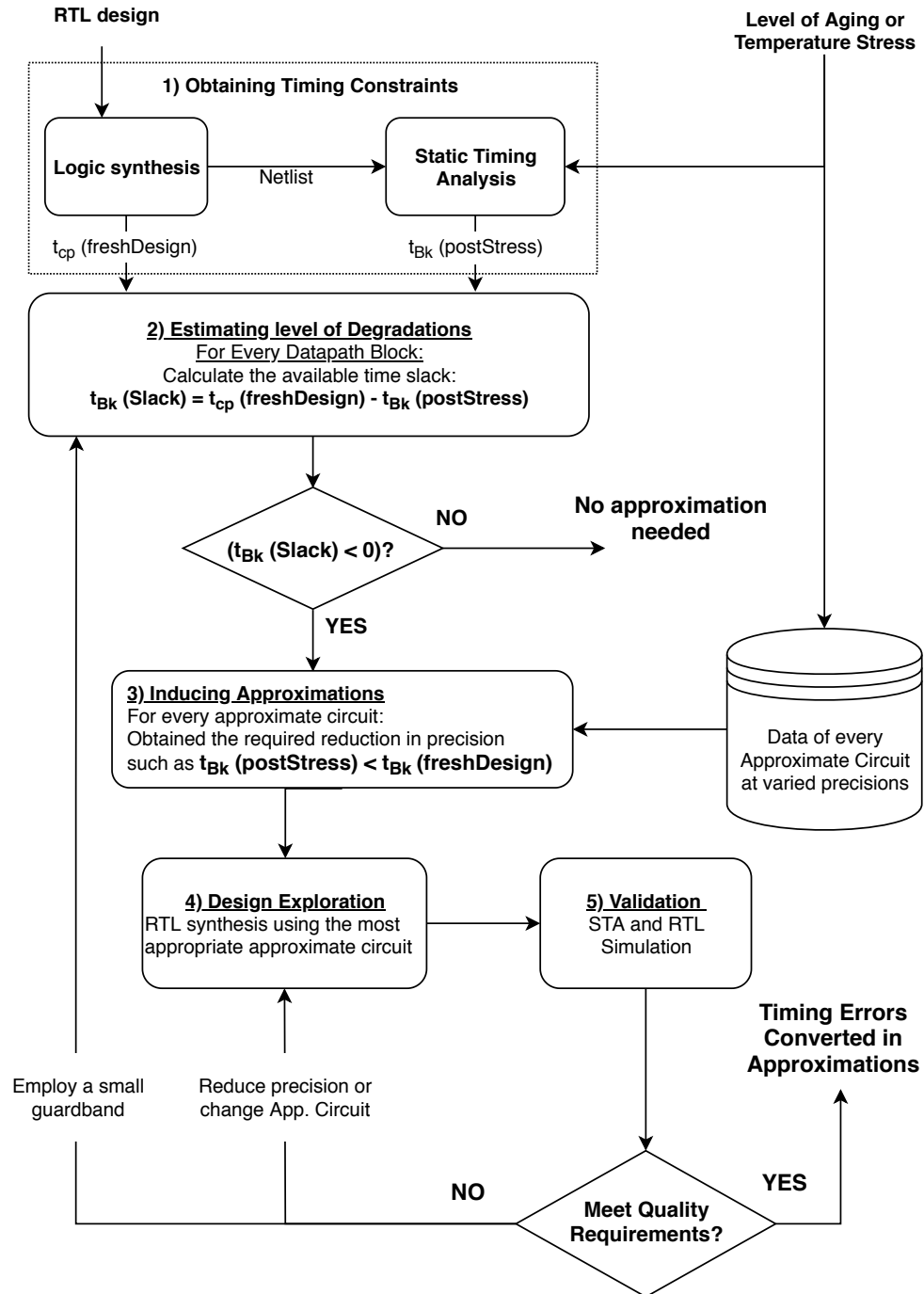


Figure 5.3: Design methodology at the architectural level to convert degradations to controllable errors using approximate circuits.

between the timing constraint and the delay of each block considering degradations ($t_{B_k}(postStress)$). While a *positive* time slack means no guard-bands are needed, a *negative* value (i.e., $t_{B_k}(slack) < 0$) means that timing violations will occur in the corresponding component. Hence, delay guardbands are required to avoid catastrophic errors, which leads to hardware performance (speed) loss. On the other hand, the negative slack can be compensated with approximations to maintain the speed in the architecture.

$$t_{B_k}(slack) = t_{CP}(freshDesign) - t_{B_k}(postStress) \quad (5.7)$$

3) Inducing Controllable Approximations

In practice, glue and steering logic exist during the process of synthesis. Therefore, the characterized components in Section 4 (see Fig. 4.1) can be employed to efficiently compensate for the existing time slack according to the target application. It is assumed that every block B_k contains an arithmetic circuit that can be approximated. Depending on how large the existing time slack is, the precision reduction can be the maximum precision reduction allowed for the approximate circuit or smaller.

4) Design Exploration

Different approximate schemes with different circuit characteristics may require a different level of precision to meet the same timing goal. Therefore, at this stage, we can choose an approximate circuit according to the output quality or circuit characteristics. After determining the most suitable approximation circuit, we implement suitable modifications in the RTL and repeat the process of synthesis to optimize

the glue logic surrounding the approximate components. By this means, we are also trading power and speed while keeping reliability.

5) Validating Timing Constraints and Output Quality

We then perform degradation-aware STA with the new netlist and a functional RTL simulation to ensure that we met the timing constraints and output quality, respectively. Note that there is a small likelihood that a small negative timing slack remains. This can be due to an increase in the degradation-induced delay in the glue logic surrounding components. In such a case, another reduction of precision to compensate for the remaining slack will be necessary. As a second option, another approximate circuit with a larger error can be investigated (by a design exploration). If the final quality output is not sufficient, we can increase the precision at the cost of a small guardband. However, such a guardband will be significantly smaller than the original one when no approximations are applied.

5.4 Experimental Setup

In the scope of this evaluation, we aimed to mitigate guard-bands for a temperature of $70^{\circ}C$ under two different design constraints. Signed multipliers are used for the IDCT application, and unsigned multipliers for the smoothing and sharpening applications. The RTL designs for these applications are synthesized with the 45-nm Nangate technology library [68] using the Synopsys Design Compiler. During the post-stress phase, we ran STA with Prime Time to obtain the maximum delay in the circuit after inducing temperature with the degradation-aware cell libraries [69]. The PSNR metric is used to evaluate the output quality of 10 representative image files during

the validation stage. In this thesis, we aimed at an output of at least 30 dB , which is commonly considered an acceptable image quality output [38]. Finally, to compare this approach against state-of-the-art guardband techniques, the three applications are synthesized with accurate circuits using the degradation-aware synthesis approach proposed in [31].

5.5 Simulation Results

The timing reports obtained from the Synopsys tools indicate that the multiplier constrains the critical path in the three image processing applications. Hence, the pre-characterized libraries in Section 4.3 can be employed to mitigate the temperature-induced delay with approximations during the design exploration stage (see methodology in Fig. 5.3). Table 4.3 (in Chapter 4) shows that truncation of 7 bits in the unsigned and signed multipliers is the best option to mitigate delay guardbands at 70°C. Following the proposed methodology, the accurate circuits are replaced by approximate circuits and then re-synthesized to optimize the surrounding glue logic. Note here that the timing constraint used during the synthesis process has to be modified to the smallest value of the new approximate circuit. Otherwise, the Design Compiler will relax the timing constraint and improve the other circuits measures rather than decreasing the critical path delay.

Table 5.1 shows the circuit measures and output quality for the three applications. It should be note that the delay column will determine the maximum clock frequency while still guarantying reliability in the circuit. However, a negative value in the slack column indicates that the design does not achieve the constrained timing. Therefore, a negative slack value can also be translated as the required delay guardband in the

Table 5.1: Measures for the high-performance applications running at $70^{\circ}C$

Application	Delay (<i>ns</i>)	Slack [†] (<i>ns</i>)	Area (<i>mm</i> ²)	Power (<i>mW</i>)	PDP (<i>pJ</i>)	PSNR ^{††} (<i>dB</i>)
IDCT:						
Accurate [*]	0.86	-0.23	27.65	31.30	26.81	43.49
TBM-7 [‡]	0.63	0.00	18.33	30.90	19.47	30.19
BBM-8 [‡]	0.78	-0.15	25.50	34.00	26.52	31.64
Smoothing:						
Accurate [*]	0.84	-0.21	3.64	2.98	2.50	∞
TruM-7 [‡]	0.63	0.00	1.98	2.14	1.34	6.76
TAM1-16 [‡]	0.64	-0.01	2.41	2.36	1.50	37.27
Sharpening:						
Accurate [*]	0.87	-0.24	4.50	3.50	3.04	∞
TruM-7 [‡]	0.65	-0.02	2.40	2.62	1.71	18.56
TAM1-16 [‡]	0.66	-0.03	2.85	2.92	1.91	45.56

[†] The required time is defined by the circuit without any degradations.

^{††} Average of ten different images commonly found in multimedia applications.

^{*} The RTL implementation employs degradation-aware synthesis [31].

[‡] The RTL implementation employs our degradation-induced approximation.

circuit to avoid timing violations. Regarding the IDCT application, simulation results not only indicate that the TBM-7 meets the requirement of 30 *dB*, but also this approximate scheme completely remove delay guardbands with positive gains in performance compared with the accurate circuit using the degradation-aware synthesis. Interestingly, we found that TruM-7 considerably degrades the output quality down to 6.76 *dB* and 18.56 *dB* for the smoothing and sharpening applications, respectively. Considering that performance is the most critical aspect for these architectures, we explored other approximate circuits prior to increasing the precision of TruM at the cost of small guardbands. In this process, we found that the TAM1-16 [58] improves the output quality significantly while still meeting the delay requirement. Although

Table 5.2: Measures for the low-power applications running at 70°C

Application	Delay (<i>ns</i>)	Slack [†] (<i>ns</i>)	Area (<i>mm</i> ²)	Power (<i>mW</i>)	PDP (<i>pJ</i>)	PSNR ^{††} (<i>dB</i>)
IDCT:						
Accurate*	2.46	-0.52	24.68	10.90	26.78	43.49
TBM-7‡	1.58	0.36	16.69	11.60	18.38	30.19
BBM-8‡	2.05	-0.11	24.00	13.00	26.61	31.64
Smoothing:						
Accurate*	2.50	-0.44	2.63	0.77	1.93	∞
TruM-5‡	2.17	-0.11	1.60	0.56	1.21	17.10
TAM2-16‡	2.06	0.00	1.73	0.62	1.27	37.17
Sharpening:						
Accurate*	2.49	-0.40	3.58	1.01	2.51	∞
TruM-5‡	2.13	-0.04	2.28	0.78	1.66	35.62
TAM2-16‡	2.07	0.02	2.28	0.75	1.55	46.44

[†] The required time is defined by the circuit without any degradations.

^{††} Average of ten different images commonly found in multimedia applications.

* The RTL implementation employs degradation-aware synthesis [31].

‡ The RTL implementation employs our degradation-induced approximation.

these gains in output quality come at the expenses of an increase in power compared to TruM-7, we still observed overall positive gains (delay, area and power) compared to the degradation-aware synthesis methodology (see section 2.2) [31].

In the second experiment, we repeated the process but then considering a different design constraint (low-power) for the same RTL applications. Despite the delay guard-bands that could be mitigated by using faster logic gates in low-power architectures, this approach leads to higher area and power consumption. Our approximate approach consists of maintaining the performance without affecting other circuit metrics. Table 5.2 shows the simulation results for the three image processing applications synthesized for low-power. Similar to the high-performance applications, the TBM-7

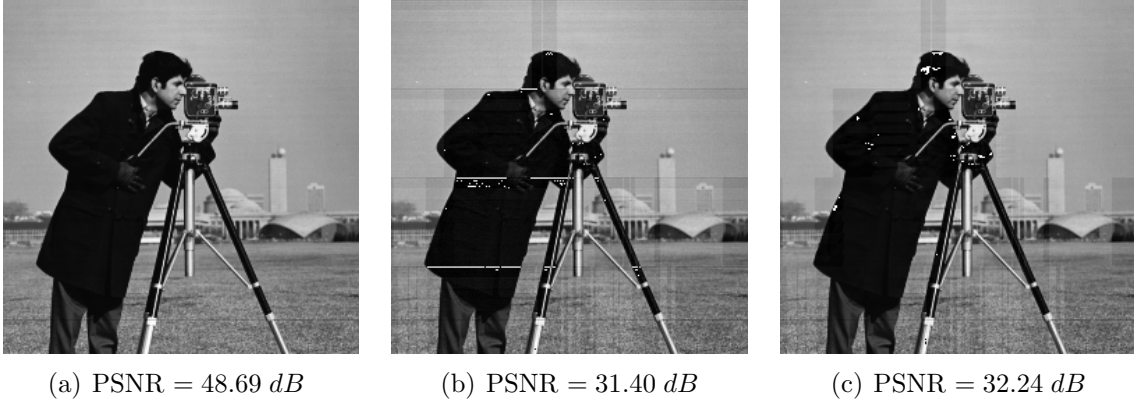


Figure 5.4: IDCT outputs when the chip is exposed at 70°C using: (a) degradation-aware synthesis with an accurate circuit, (b) our approximate approach with TBM-7, and (c) our approximate approach with BBM-8.

meets the requirement of 30 dB and the timing goal for the IDCT architecture. Regarding the sharpening and smoothing applications, TruM-5 design has the lowest MSE towards temperature-induced delay at 70°C for the unsigned multipliers (see Table 4.4). However, the final output quality of the image processing applications is extremely low with this approximate scheme. Although the TAM2-16 and TruM-5 show similar characteristics in terms of circuit metrics at the architecture level, TAM2-16 generates a better output quality especially in the smoothing application.

Figures 5.4, 5.5 and 5.6 show a visual comparison of the output images for the IDCT, image smoothing and sharpening applications, respectively. As mentioned, the main objective for this methodology is to accurately trade-off delay guardbands for a loss in output quality. Despite degradation aware synthesis shows the best quality output employing accurate circuits, this approach incurs a penalty in hardware performance with the addition of delay guard-bands. On the other hand, the proposed methodology in this section not only mitigates delay guard-bands with a minimum reduction in the output quality, but also improve other circuit metrics such as area

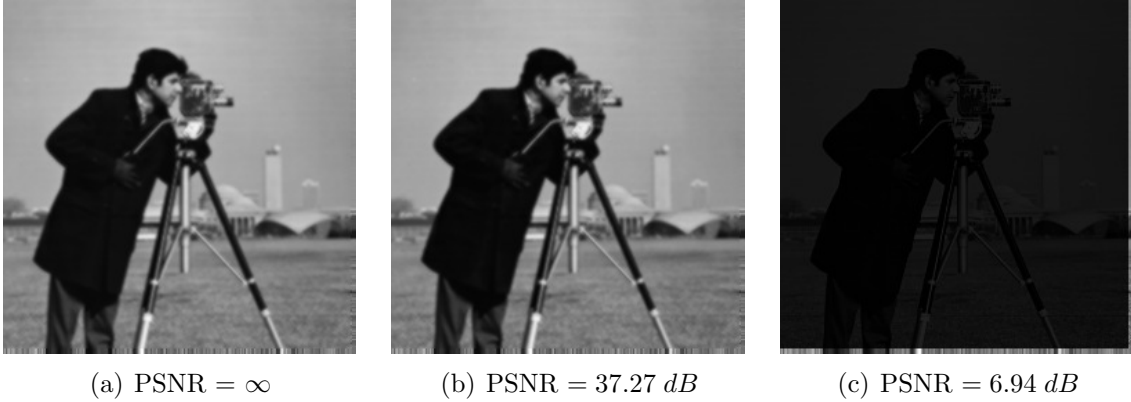


Figure 5.5: Image smoothing outputs when the chip is exposed at 70°C using: (a) degradation-aware synthesis with an accurate circuit, (b) our approximate approach with TAM1-16, and (c) our approximate approach with TruM-7.

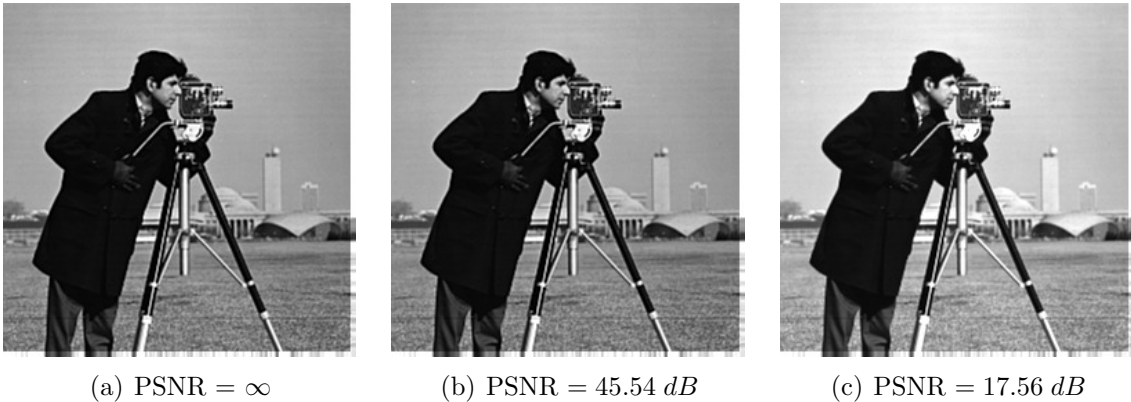


Figure 5.6: Image sharpening outputs when the chip is exposed at 70°C using: (a) degradation-aware synthesis with an accurate circuit, (b) our approximate approach with TAM1-16, and (c) our approximate approach with TruM-7.

and power efficiency.

5.6 Summary

This chapter discusses a complete framework to mitigate or completely remove guard-bands at the architectural level using the principles of approximate computing. The

presented framework was applied to overcome temperature-induced degradations at $70^{\circ}C$ without degrading the performance when using delay guard-bands. Although TruM shows a lower MSE than TAM1 and TAM2 at the component level, the simulation results show that TAM1 and TAM2 provide better output quality (with a higher PSNR) than truncation of LSBs at the architectural level. On the other hand, TruM shows a larger MRED than TAM1 and TAM2 at the component level. Therefore, the MRED can be considered as a better error metric to predict the output quality of the studied image processing applications. Finally, the results in Tables 5.1 and 5.2 show that the accurate multipliers in three different image processing applications can be replaced by approximate multipliers for guardband mitigation purposes at the cost of negligible image quality degradation.

Chapter 6

Conclusions

6.1 Summary

Aggressive scaling has reached a limit where certain aspects endanger the correct functionality of CMOS circuits. Among multiple factors, the aging phenomena and variations in chip temperature became the main concern in terms of reliability. Therefore, accurately exploring the impact of transistor degradations is a prerequisite even for the design of error-tolerant applications.

Approximate arithmetic circuits have brought significant gains in performance and power. In Chapter 3, traditional manual and automated search-based approximate circuit designs are reviewed, evaluated, and compared. Different from the current literature, the performance impact that occurred due to different levels of degradations is considered during the process of evaluation. First, we show how the functionality of the approximate circuits varies based on degradation-induced timing violations. The approximate schemes that are primarily designed to reduce the critical path are more resilient to degradations. For example, the segmented adders, even though this

scheme has multiple critical paths, become the most reliable approximate designs during run-time if a timing violation occurs. On the contrary, a larger critical path means a more significant level of degradation. Consequently, less time remains to perform the computations in a larger number of MSBs and thus incurring a larger error. In this chapter, we also compare the performance of the approximate circuits using a state-of-the-art guard-band technique to avoid the time violation in the circuits. Although we found cases in which the critical path in the absence of degradations does not remain as the critical path in an approximate circuit after circuit degradations, our simulation results indicate that the required guard-bands to overcome degradations for this approximate circuit does not make it better or worse in terms of speed compared to other approximate designs with their respective delay guard-bands. Therefore, we can guarantee that the selection of one approximate circuit without employing a technique to sustain lifetime reliability will be the best design after applying a guard-band technique to sustain circuit lifetime reliability. This is an important finding as it would not be necessary to evaluate all the approximate circuits under different levels of work conditions to determine optimal solutions according to the application requirements, although the proposed simulation methodology can be used to obtain quantitative measures of the circuit performance.

Chapter 4 discusses a methodology to accurately convert degradations into controllable errors at the component level. Despite the fact that this methodology has been broadly used in the literature for synthesis and adaptive techniques, only the truncation of LSBs has been considered. The novelty in this thesis work lies in the evaluation of a large number of approximate arithmetic circuits to determine the

optimal solution under different application requirements. The conducted experiments show that a truncated adder is not the most effective technique, although it has been extensively used in the current literature. Among all the approximate adders, automatically generated adders using CGP produce the lowest error when we aimed to mitigate small degradations (aging-in/duced timing errors) followed by LOAs. Most of the speculative adders are designed for a high-speed by cutting the carry chain, which makes them suitable to overcome larger degradations such as high-temperatures. Of the considered approximate multipliers, the truncated multiplier is the most effective design methodology with respect to MSE. However, AM2, AM1 and TAM2 are the most effective approximate schemes by considering the MRED.

In Chapter 5, we effectively convert degradations at the application level based on the pre-characterization of degradations at the component level. Three different image processing applications have been evaluated in the context of this thesis work. The conducted experiments showed that temperature-induced degradation leads to an unacceptable quality loss, even for error-tolerant applications. Compared with the designs in literature, different approximation techniques were explored in more detail to trade-off guard-bands for approximations. The simulation results show that the MRED obtained at the component level is more relevant to application-specific metrics such as the PSNR. In the context of signed multipliers, we demonstrated that guard-bands are not only completely removed towards temperature-induced approximation, but also a gain of 28% in the PDP is achieved compared with the state-of-the-art approach from [31]. Similarly, we demonstrated that the TAM1 and TAM2 are the most effective schemes towards guardband mitigation for the high-performance and low-power applications, respectively.

Bibliography

- [1] M. Alioto, “Enabling the Internet of Things: From integrated circuits to integrated systems,” in *Springer Publishing Company, Incorporated*, 2017.
- [2] S. Borkar and A. A. Chien, “The future of microprocessors,” *Commun. ACM*, vol. 54, no. 5, pp. 67–77, May 2011. [Online]. Available: <http://doi.acm.org/10.1145/1941487.1941507>
- [3] A. Agarwal, C. H. Kim, S. Mukhopadhyay, and K. Roy, “Leakage in nano-scale technologies: Mechanisms, impact and design considerations,” in *Proceedings of the 41st Annual Design Automation Conference*, ser. DAC '04. ACM, 2004, pp. 6–11.
- [4] H. Amrouch, B. Khaleghi, and J. Henkel, “Optimizing temperature guardbands,” in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 175–180.
- [5] M. Shafique, S. Garg, J. Henkel, and D. Marculescu, “The EDA challenges in the dark silicon era,” in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2014, pp. 1–6.

- [6] J. M. Cardoso, J. G. F. Coutinho, and P. C. Diniz, “Chapter 2 - high-performance embedded computing,” in *Embedded Computing for High Performance*. Morgan Kaufmann, 2017, pp. 17 – 56.
- [7] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, “Analysis and characterization of inherent application resilience for approximate computing,” in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2013, pp. 1–9.
- [8] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in *2013 18th IEEE European Test Symposium (ETS)*, 2013, pp. 1–6.
- [9] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, “Approximate arithmetic circuits: A survey, characterization and recent applications,” in *Proceedings of the IEEE 2020*, 2020.
- [10] G. Karakonstantis, D. Mohapatra, and K. Roy, “System level DSP synthesis using voltage overscaling, unequal error protection adaptive quality tuning,” in *2009 IEEE Workshop on Signal Processing Systems*, Oct 2009, pp. 133–138.
- [11] R. Hegde and N. R. Shanbhag, “Soft digital signal processing,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 6, pp. 813–823, Dec 2001.
- [12] S. Reda and M. Shafique, *Approximate Circuits: Methodologies and CAD*. Springer International Publishing, 2018. [Online]. Available: <https://books.google.ca/books?id=Drh9DwAAQBAJ>

-
- [13] H. Jiang, “Design, evaluation and application of approximate arithmetic circuits,” Ph.D. dissertation, University of Alberta, 2018.
- [14] A. Alaghi and J. P. Hayes, “Survey of stochastic computing,” *ACM Trans. Embed. Comput. Syst.*, vol. 12, no. 2s, pp. 92:1–92:19, May 2013.
- [15] S. T. Chakradhar and A. Raghunathan, “Best-effort computing: Re-thinking parallel software and hardware,” in *Design Automation Conference*, June 2010, pp. 865–870.
- [16] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, “Managing performance vs. accuracy trade-offs with loop perforation,” in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, ser. ESEC/FSE ’11, 2011, pp. 124–134.
- [17] W. Baek and T. M. Chilimbi, “Green: A framework for supporting energy-conscious programming using controlled approximation,” in *Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI ’10. New York, NY, USA: ACM, 2010, pp. 198–209. [Online]. Available: <http://doi.acm.org/10.1145/1806596.1806620>
- [18] H. Amrouch, “Techniques for aging, soft errors and temperature to increase the reliability of embedded on-chip systems,” Ph.D. dissertation, Karlsruhe Institute of Technology, 2015.
- [19] G. Moore, “The future of integrated electronics,” in *Fairchild Semiconductor internal publication*, 1964.

- [20] H. Amrouch and J. Henkel, "Containing guardbands," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2017, pp. 537–542.
- [21] S. Mittal, "A survey of architectural techniques for managing process variation," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 54:1–54:29, Feb. 2016. [Online]. Available: <http://doi.acm.org/10.1145/2871167>
- [22] M. A. Scarpato, "Digital circuit performance estimation under PVT and aging effects," Ph.D. dissertation, Université Grenoble Alpes, 2017.
- [23] S. K. Saha, "Modeling process variability in scaled CMOS technology," *IEEE Design Test of Computers*, vol. 27, no. 2, pp. 8–16, March 2010.
- [24] N. A. Drego, "A low-skew, low-jitter receiver circuit for on-chip optical clock distribution," Master's thesis, University of California, Irvine, 2001.
- [25] M. Wirnshofer, *Variation-Aware Adaptive Voltage Scaling for Digital CMOS Circuits*. Springer Publishing Company, Incorporated, 2013.
- [26] "Definition of: High-k/metal gate," accessed: 09 January 2020. [Online]. Available: <http://www.pcmag.com/encyclopedia/term/58937/high-k-metal-gate>
- [27] V. M. van Santen, H. Amrouch, and J. Henkel, "New worst-case timing for standard cells under aging effects," *IEEE Transactions on Device and Materials Reliability*, vol. 19, no. 1, pp. 149–158, March 2019.
- [28] "Synopsys EDA tools," accessed: 2019-06-28. [Online]. Available: <http://www.synopsys.com/>

- [29] M. Ebrahimi, F. Oboril, S. Kiamehr, and M. B. Tahoori, "Aging-aware logic synthesis," in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2013, pp. 61–68.
- [30] 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 10, pp. 1618–1629, Oct 2016.
- [31] H. Amrouch, B. Khaleghi, A. Gerstlauer, and J. Henkel, "Reliability-aware design to suppress aging," in *2016 53rd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2016, pp. 1–6.
- [32] H. Amrouch, S. Mishra, V. van Santen, S. Mahapatra, and J. Henkel, "Impact of BTI on dynamic and static power: From the physical to circuit level," in *2017 IEEE International Reliability Physics Symposium (IRPS)*, April 2017, pp. CR-3.1–CR-3.6.
- [33] L. Zhang and R. P. Dick, "Scheduled voltage scaling for increasing lifetime in the presence of NBTI," in *2009 Asia and South Pacific Design Automation Conference*, Jan 2009, pp. 492–497.
- [34] S. Das, C. Tokunaga, S. Pant, W. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RazorII: In situ error detection and correction for PVT and SER tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan 2009.

- [35] M. Sadi, G. K. Contreras, J. Chen, L. Winemberg, and M. Tehranipoor, "Design of reliable SoCs with BIST hardware and machine learning," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 11, pp. 3237–3250, Nov 2017.
- [36] K. Huang, X. Zhang, and N. Karimi, "Real-time prediction for ic aging based on machine learning," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–9, 2019.
- [37] D. Palomino, M. Shafique, A. Susin, and J. Henkel, "Thermal optimization using adaptive approximate computing for video coding," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 1207–1212.
- [38] H. Amrouch, B. Khaleghi, A. Gerstlauer, and J. Henkel, "Towards aging-induced approximations," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2017, pp. 1–6.
- [39] B. Boroujerdian, H. Amrouch, J. Henkel, and A. Gerstlauer, "Trading off temperature guardbands via adaptive approximations," in *2018 IEEE 36th International Conference on Computer Design (ICCD)*, Oct 2018, pp. 202–209.
- [40] J. Kim, H. Kim, H. Amrouch, J. Henkel, A. Gerstlauer, and K. Choi, "Aging gracefully with approximation," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2019, pp. 1–5.
- [41] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, "A review, classification, and comparative evaluation of approximate arithmetic circuits," *J. Emerg. Technol.*

- Comput. Syst.*, vol. 13, no. 4, pp. 60:1–60:34, Aug. 2017. [Online]. Available: <http://doi.acm.org/10.1145/3094124>
- [42] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, “Design of voltage-scalable meta-functions for approximate computing,” in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.
- [43] Ning Zhu, W. L. Goh, and K. S. Yeo, “An enhanced low-power high-speed adder for error-tolerant application,” in *Proceedings of the 2009 12th International Symposium on Integrated Circuits*, Dec 2009, pp. 69–72.
- [44] Y. Kim, Y. Zhang, and P. Li, “Energy efficient approximate arithmetic for error resilient neuromorphic computing,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2733–2737, Nov 2015.
- [45] J. Hu and W. Qian, “A new approximate adder with low relative error and correct sign calculation,” in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2015, pp. 1449–1454.
- [46] I. Lin, Y. Yang, and C. Lin, “High-performance low-power carry speculative addition with variable latency,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1591–1603, Sep. 2015.
- [47] K. Du, P. Varman, and K. Mohanram, “High performance reliable variable latency carry select addition,” in *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2012, pp. 1257–1262.
- [48] L. Li and H. Zhou, “On error modeling and analysis of approximate adders,” *ICCAD*, pp. 511–518, 2014.

- [49] A. K. Verma, P. Brisk, and P. Ienne, “Variable latency speculative addition: A new paradigm for arithmetic circuit design,” in *2008 Design, Automation and Test in Europe*, March 2008, pp. 1250–1255.
- [50] A. B. Kahng and S. Kang, “Accuracy-configurable adder for approximate arithmetic designs,” in *DAC Design Automation Conference 2012*, June 2012, pp. 820–825.
- [51] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 850–862, April 2010.
- [52] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, “Design-efficient approximate multiplication circuits through partial product perforation,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 10, pp. 3105–3117, Oct 2016.
- [53] P. Kulkarni, P. Gupta, and M. Ercegovac, “Trading accuracy for power with an underdesigned multiplier architecture,” in *2011 24th International Conference on VLSI Design*, Jan 2011, pp. 346–351.
- [54] K. Y. Kyaw, W. L. Goh, and K. S. Yeo, “Low-power high-speed multiplier for error-tolerant application,” in *2010 IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC)*, Dec 2010, pp. 1–4.

- [55] C. Lin and I. Lin, "High accuracy approximate multiplier with error correction," in *2013 IEEE 31st International Conference on Computer Design (ICCD)*, Oct 2013, pp. 33–38.
- [56] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, April 2015.
- [57] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2014, pp. 1–4.
- [58] H. Jiang, C. Liu, F. Lombardi, and J. Han, "Low-power approximate unsigned multipliers with configurable error recovery," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 189–202, Jan 2019.
- [59] F. Farshchi, M. S. Abrishami, and S. M. Fakhraie, "New approximate multiplier for low power digital signal processing," in *The 17th CSI International Symposium on Computer Architecture Digital Systems (CADSD 2013)*, Oct 2013, pp. 25–30.
- [60] L. Sekanina, Z. Vasicek, and V. Mrazek, *Automated Search-Based Functional Approximation for Digital Circuits*. Springer International Publishing, 2019, pp. 175–203. [Online]. Available: https://doi.org/10.1007/978-3-319-99322-5_9
- [61] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "Evoapprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of

- approximation methods,” in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 258–261.
- [62] V. Mrazek, Z. Vasicek, L. Sekanina, H. Jiang, and J. Han, “Scalable construction of approximate multipliers with formally guaranteed worst case error,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 11, pp. 2572–2576, Nov 2018.
- [63] J. Liang, J. Han, and F. Lombardi, “New metrics for the reliability of approximate and probabilistic adders,” *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760–1771, Sep. 2013.
- [64] C. Liu, J. Han, and F. Lombardi, “An analytical framework for evaluating the error characteristics of approximate adders,” *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1268–1281, May 2015.
- [65] Y. Wu, Y. Li, X. Ge, Y. Gao, and W. Qian, “An efficient method for calculating the error statistics of block-based approximate adders,” *IEEE Transactions on Computers*, vol. 68, no. 1, pp. 21–38, Jan 2019.
- [66] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, “Macaco: Modeling and analysis of circuits for approximate computing,” in *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2011, pp. 667–673.
- [67] H. Jiang, F. J. H. Santiago, M. S. Ansari, L. Liu, B. F. Cockburn, F. Lombardi, and J. Han, “Characterizing approximate adders and multipliers optimized under

- different design constraints,” in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, 2019, p. 393–398.
- [68] “Nangate, Open Cell Library.” [Online]. Available: <http://www.nangate.com/>
- [69] “Degradation-Aware Cell Libraries, V1.0.” [Online]. Available: <http://ces.itec.kit.edu/dependable-hardware.php>
- [70] V. Chandra, “Monitoring reliability in embedded processors - a multi-layer view,” in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2014, pp. 1–6.
- [71] J. Bhasker and R. Chadha, “Static timing analysis for nanometer designs: A practical approach,” *Springer*, 2009.
- [72] A. S. Mutschler, “Thoroughly verifying complex SoCs,” accessed: 08 January 2020. [Online]. Available: <https://semiengineering.com/thoroughly-verifying-complex-socs/>
- [73] “The DCT/IDCT solution customer tutorial,” accessed: 2019-Dec-18. [Online]. Available: www.xilinx.com
- [74] H. R. Myler and A. R. Weeks, *The Pocket Handbook of Image Processing Algorithms in C*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.
- [75] M. S. Lau, K.-V. Ling, and Y.-C. Chu, “Energy-aware probabilistic multiplier: Design and analysis,” in *Proceedings of the 2009 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, ser. CASES

'09. New York, NY, USA: ACM, 2009, pp. 281–290. [Online]. Available:
<http://doi.acm.org/10.1145/1629395.1629434>