UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

GUILHERME PEREIRA PAIM

Approximate and Timing-Speculative Hardware Design for High-Performance and Energy-Efficient Video Processing

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Microeletronics

Advisor: Prof. Dr. Sergio Bampi
Coadvisor: Prof. Dr. Eduardo A. C. Costa

Porto Alegre
March 2021

CIP — CATALOGING-IN-PUBLICATION

Pereira Paim, Guilherme

Approximate and Timing-Speculative Hardware Design for High-Performance and Energy-Efficient Video Processing / Guilherme Pereira Paim. – Porto Alegre: PGMICRO da UFRGS, 2021.

158 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica, Porto Alegre, BR–RS, 2021. Advisor: Sergio Bampi; Coadvisor: Eduardo A. C. Costa.

1. Approximate Computing. 2. Timing-Speculative. 3. High-Performance. 4. Energy-Efficient. 5. Voltage Over-Scaling. 6. Temperature Rising. 7. Device Aging. 8. Video Processing. I. Bampi, Sergio. II. Costa, Eduardo A. C.. III. Título.

“The fear of falling can’t be greater than the passion to fly“

— FILIPE RET - BRAZILIAN RAPPER

“Learn the rules like a Pro, so you can break them like an artist.“

Attributed to — PABLO PICASSO

AGRADECIMENTOS

Agradeço primeiramente aos brasileiros que apoiam a ciência e tecnologia nacional por meio das agências CAPES e CNPq que financiaram parte da minha formação.

Agradeço aos meus pais e avós pelo seu amor, dedicação e apoio incondicional.

Agradeço à minha companheira Ana Karina Christ, por todo o seu carinho, apoio e motivação. Agradeço à Ana Karina especialmente por ter me ajudado a persistir durante a pandemia, um momento sombrio, especialmente para o Brasil. Parabenizo a Ana Karina também por ter recentemente conquistado a valiosa oportunidade de desenvolver o seu doutorado no Instituto Superior Técnico da Universidade de Lisboa.

Gostaria de agradecer especialmente ao grande amigo e co-orientador, Prof. Eduardo Costa. Agradeço ao Prof. Eduardo por ter me motivado, contribuído e participado ativamente em todos os momentos da minha formação, desde a minha graduação, ingresso no doutorado, no doutorado sanduíche na Alemanha, e nos papers que trabalhamos duro juntos, até agora durante esta etapa final do desenvolvimento desta tese. Quero agradecer por sua grande parceria, presença nos copos, imperiais e finos, e também quero extender este agradecimento à sua companheira Nicácia. Quero agradecer também ao Prof. Sergio Almeida da UCPel pela amizade, parceria e contribuições ao meu doutorado.

Gostaria de agradecer ao meu brilhante orientador e grande amigo, Prof. Sergio Bampi. O Prof. Bampi é um motivo de inspiração, de trajetória, luta e compromisso com o desenvolvimento da academia e da indústria de semicondutores no Brasil. Agradeço à ele pela sua excelência na minha orientação e pelo imenso apoio quando eu estava no exterior. Este apoio que somado ao do Prof. Eduardo, foi fundamental para o excelente sucesso do meu doutorado sanduíche junto ao KIT na Alemanha.

Quero aqui deixar registrado meu agradecimento ao grupo do Lab. 215, do prédio 67, do Campus do Vale da UFRGS. Agradeço ao Dr. Leandro Rocha, meu primo, grande amigo, e ilustre colega do Lab. 215, pela brilhante e simbiótica dupla que ele fez comigo durante este doutorado, *to infinity & beyond*. Agradeço também ao notável colega e grande amigo Brunno Abreu, *Goldenboy*, pela parceria no Lab. 215, pelas suas valiosas contribuições técnico-científicas nos trabalhos e as aventuras pelas conferências nos USA e na Europa. Agradeço à outra lenda-viva do Lab. 215, meu amigo Gustavo Santana, e parabenizo-o também pelo seu ingresso ao doutorado na Yale University. Aos colegas da antiga do Lab. 215, agradeço aos amigos Dr. Mateus Grellert, Dr. Léo Soares, Dra. Duda Monteiro e Dr. Dieison Silveira pelas várias discussões técnicas e outras de pura zoeira.

I am thankful to my advisor in Germany, Prof. Henkel, for his hospitality in receiving me in the CES (Chair for Embedded Systems) at KIT (Karlsruhe Institute of Technology). I want to thank my supervisor in Germany, Prof. Hussam Amrouch, for which the partnership, technical and scientific advising were highly valuable to this thesis development. I am thankful to Dr. Georgios Zervakis for his partnership and co-authorship, revisions in other works, and his ideas that improve my framework implementation. I want to thank my Indian legendary friends of the German Lab, Dr. Om Prakash, Govind Bajpai, and Aniket Gupta. I wanto to thank also my friends, Florian Klemme and Tanfer Alan, for their partnership in Germany, the beers, and cheers. También quiero agradecer a mi amigo el Dr. Jorge Castro-Godínez, gracias por la colaboración en el laboratorio en Alemania.

Finalmente, gostaria de agradecer aos grandes professores do PGMicro, demais colegas, professores e parceiros de luta que dedicam sua vida para que a academia e indústria de semicondutores brasileira seja continuamente considerada estratégica e prioritária, e para que recupere a posição de destaque de que gozou no passado recente.

ABSTRACT

Since the end of transistor scaling in 2-D appeared on the horizon, innovative circuit design paradigms have been on the rise to go beyond the well-established and ultra-conservative exact computing. Many compute-intensive applications – such as video processing – exhibit an intrinsic error resilience and do not necessarily require perfect accuracy in their numerical operations. Approximate computing (AxC) is emerging as a design alternative to improve the performance and energy-efficiency requirements for many applications by trading its intrinsic error tolerance with algorithm and circuit efficiency. Exact computing also imposes a worst-case timing to the conventional design of hardware accelerators to ensure reliability, leading to an efficiency loss. Conversely, the timing-speculative (TS) hardware design paradigm allows increasing the frequency or decreasing the voltage beyond the limits determined by static timing analysis (STA), thereby narrowing pessimistic safety margins that conventional design methods implement to prevent hardware timing errors. Timing errors should be evaluated by an accurate gate-level simulation, but a significant gap remains: How these timing errors propagate from the underlying hardware all the way up to the entire algorithm behavior, where they just may degrade the performance and quality of service of the application at stake? This thesis tackles this issue by developing and demonstrating a cross-layer framework capable of performing investigations of both AxC (i.e., from approximate arithmetic operators, approximate synthesis, gate-level pruning) and TS hardware design (i.e., from voltage over-scaling, frequency over-clocking, temperature rising, and device aging). The cross-layer framework can simulate both timing errors and logic errors at the gate-level by crossing them dynamically, linking the hardware result with the algorithm-level, and vice versa during the evolution of the application's runtime. Existing frameworks perform investigations of AxC and TS techniques at circuit-level (i.e., at the output of the accelerator) agnostic to the ultimate impact at the application level (i.e., where the impact is truly manifested), leading to less optimization. Unlike state of the art, the framework proposed offers a holistic approach to assessing the tradeoff of AxC and TS techniques at the application-level. This framework maximizes energy efficiency and performance by identifying the maximum approximation levels at the application level to fulfill the required good enough quality. This thesis evaluates the framework with an 8-way SAD (Sum of Absolute Differences) hardware accelerator operating into an HEVC encoder as a case study. Application-level results showed that the SAD based on the approximate adders achieve savings of up to 45% of energy/operation with an increase of only 1.9% in BD-BR. On the other hand, VOS (Voltage Over-Scaling) applied to the SAD generates savings of up to 16.5% in energy/operation with around 6% of increase in BD-BR. The framework also reveals that the boost of about 6.96% (at 50°) to 17.41% (at 75° with 10-Y aging) in the maximum clock frequency achieved with TS hardware design is totally lost by the processing overhead from 8.06% to 46.96% when choosing an unreliable algorithm to the blocking match algorithm (BMA). We also show that the overhead can be avoided by adopting a reliable BMA. This thesis also shows approximate DTT (Discrete Tchebichef Transform) hardware proposals by exploring a transform matrix approximation, truncation, and pruning. The results show that the approximate DTT hardware proposal increases the maximum frequency up to 64%, minimizes the circuit area in up to 43.6%, and saves up to 65.4% in power dissipation. The DTT proposal mapped for FPGA shows an increase of up to 58.9% on the maximum frequency and savings of about 28.7% and 32.2% on slices and dynamic power, respectively, compared with state of the art.

Keywords: Approximate Computing. Timing-Speculative. High-Performance. Energy-Efficient. Voltage Over-Scaling. Temperature Rising. Device Aging. Video Processing.

LIST OF ABBREVIATIONS AND ACRONYMS

AA	Approximate Adder
ACA	Almost Correct Adder
AMP	Asymmetric Motion Partitions
AVC	Advanced Video Coding
ASIC	Application Specific Integrated Circuit
ASIP	Application Specific Instruction Set Processor
AxC	Approximate Computing
BD-BR	Bjøntegaard Delta Bit Rate
BMA	Block Matching Algorithm
BSIM-CMG	Berkeley Short-channel IGFET Model – Common Multi-Gate
BTI	Bias Temperature Instability
CABAC	Context Adaptive Binary Arithmetic Coding
CLA	Carry-Look Ahead
CMOS	Complementary Metal Oxide Semiconductor
CPA	Copy Adder
CPU	Central Processing Unit
CSA	Carry-Save Adder
CSLA	Carry-Select Adder
CTU	Coding Tree Unit
CU	Coding Unit
DBF	Deblocking Filter
DC	Direct Current
DCT	Discrete Cosine Transform
DPI-C	Diamond Search

DSP	Digital Signal Processing
DST	Discrete Sine Transform
DS	Diamond Search
DPI-C	Direct Programming Interface - C/C++
ETA-I	Error Tolerant Adder
FA	Full Adder
FIR	Finite Impulse Response
FME	Fractional Motion Estimation
FPGA	Field Programmable Gate Array
fps	frames per second
FS	Full Search
FHD	Full High Definition
GLS	Gate Level Simulation
GPP	General Purpose Processor
GPU	Graphics Processing Unit
HD	High-Definition
HEVC	High Efficiency Video Coding
HCID	Hot Carrier Induced Degradation
HM	HEVC Test Model Reference Software
HS	Hexagon Search
HT	Hadamard Transform
IME	Integer Motion Estimation
IoT	Internet of Things
IP	Internet Protocol
IDCT	Inverse Discrete Cosine Transform
IQ	Inverse Quantization

JCT-VC	Joint Collaborative Team on Video Coding
JM	H.264/AVC Test Model Reference Software
JPEG	Joint Photographic Experts Group
kbps	kilo bits per second
LEF	Library Exchange Format
LG	Lucky Goldstar
LOA	Lower-part-OR Adder
LSB	Least Significant Bit
NMOS	Negative-channel metal-oxide semiconductor
MAE	Mean Absolute Error
MC	Motion Compensation
MCM	Multiple Constant Multiplication
ME	Motion Estimation
MOSFET	Metal-Oxide-Semiconductor Field Effect Transistor
MSB	Most Significant Bit
MSE	Mean Square Error
PDK	Process Design Kit
PLE	Physically-Aware Layout Estimation
PMOS	Positive-channel Metal-Oxide Semiconductor
PSNR	Peek Signal-to-Noise Ratio
P2P	Peer-to-Peer Protocol
PU	Prediction Unit
QP	Quantization Parameter
Q	Quantization
RCA	Ripple Carry Adder
RDO	Rate–distortion Optimization

REF	Reference Frame
RDOQ	Rate–distortion Optimization Quantization
RGB	Red, Green, Blue
RQT	Residual Quad Tree
RTL	Register Transfer Level
SAD	Sum of Absolute Differences
SAO	Sample Adaptive Offset
SATD	Sum of Absolute Transformed Differences
SDF	Standard Delay File
SMP	Symmetric Motion Partition
SoC	System on Chip
SRAM	Static Random Access Memory
SR-SIM	Spectral Residual based Similarity
SS	Star Search
SSE	Sum of Squared Error
SSIM	Structural Similarity Index
STA	Static Timing Analysis
SV	SystemVerilog
TB	Transposition Buffer
TCAD	Technology Computer-Aided Design
TCF	Toggle Count Format
TDP	Thermal Design Power
TGB	Timing Guardbands
TU	Transform Unit
TRA	Truncation Adder
TS	Timing-Speculative

TZS	Test Zone Search
UHD	Ultra High Definition
UMHS	Uneven Multi-Hexagon Search
VCD	Value Change Dump
VOS	Voltage Over-Scaling
VHDL	Very High Speed Integrated Circuits Hardware Description Language
VLSI	Very Large Scale Integration
YCbCr	Luminance, Chrominance Blue, Chrominance Red

LIST OF SYMBOLS

$\sum$	Sum
C_L	Node Capacitance Load
f_{clk}	Operating Clock Frequency
P_{Total}	Total Power Dissipation
P_{Static}	Static Power Dissipation
V_{dd}	CMOS Voltage Supply
α	Node Switching Activity
MAX	Maximum value of the representation
$O_{i,j}$	Pixel of the Original Frame
$R_{i,j}$	Pixel of the Reference Frame
V_T	Voltage Threshold of the Transistor
μ	Carriers mobility of the Transistor

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Most advances of the impressive technological revolution saw in the last decades have been driven by microelectronics, which was intensely leveraged by the progress around the transistor's scaling. Scaling transistors have served the semiconductor industry well for more than 50 years in providing denser, cheaper, faster, and most energy-efficient integrated circuits (Bohr, 2018). Since the end of both Dennard's and transistor's scaling appeared on the horizon (Dennard, 2015), approximate design paradigms have emerging facing resistance by going against the well-established and ultra-conservative exact computing (JIAO et al., 2017). Approximate Computing (AxC) emerges as a promising paradigm to leverage the performance and/or energy efficiency of a large number of application domains by reducing the accuracy of the performed computations (CHIPPA et al., 2013). Many compute-intensive applications – such as image and video processing – exhibit an intrinsic error resilience and do not mandatorily require perfect accuracy in their numerical operations (CHIPPA et al., 2013). It has been demonstrated that such applications can tolerate errors that occurred in their underlying computations (e.g., errors originating by approximate hardware accelerators) and still deliver a satisfactory service quality (EL-HAROUNI et al., 2017).

Exact computing also imposes a worst-case timing to the conventional design of hardware accelerators to ensure reliability, leading to an efficiency loss (JIAO et al., 2017). Adversely, Timing-Speculative (TS) hardware design allows increasing the frequency or decreasing the voltage beyond the limits determined by Static Timing Analysis (STA), thereby narrowing pessimistic safety margins that conventional design implement to prevent timing errors (ASSARE; GUPTA, 2019). Voltage Over-Scaling (VOS) is an example of a widely used timing speculative technique that eases the precision requirements from the perspective of timing (ZERVAKIS et al., 2018). VOS is applied to any circuit class and consists of keeping the operational clock frequency constant and decreasing the supply voltage below its nominal value (ZERVAKIS et al., 2018).

Approximations should be carefully applied, in a disciplined manner, delivering accuracy guarantees (ESMAEILZADEH et al., 2012). Most of the related work (as in (ZERVAKIS et al., 2018; ZERVAKIS et al., 2019)) are limited to examine the errors induced by AxC and TS hardware design only at the output of the hardware accelerators (i.e., in the standalone operation). Their work is agnostic to the application considering that the error threshold can be decided in a posterior step. Nevertheless, maximizing the approximation benefits yet fulfilling enough accuracy demands a holistic approach considering the ultimate impact at the application-level. The following section demonstrates the challenges in holistically evaluate AxC and TS techniques when designing hardware accelerators targeting real-world applications. Then, we define the research question and the objectives of this thesis.

1.1 Challenges, Problem Definition and the Objectives

A key challenge of the AxC and TS hardware design is that a significant part of industrial-strength algorithms (like online machine learning, adaptive filtering, and video encoding) has a dynamic behavior embodied in the interplay between hardware blocks, accelerators, and the algorithms. Essentially, there are one or more closed-loops[1] in which the current output is dependent on the previous one. Such dynamics impose the simulation of the hardware accelerator with gate-level accuracy to capture timing errors and hence decide whether the final degradation on the overall application running will be tolerable or not. For that, the CMOS (Complementary Metal Oxide Semiconductor) gate-level simulations have to proceed at the runtime of the algorithms to obtain the delay increases resulting from TS, which trigger timing errors. Hence, this work bridges the wide gap between the high abstraction levels (system/algorithm-level) and the low abstraction levels (gate-level), towards considering the impact of AxC and TS in the hardware accelerators on the performance of the entire application dynamically, as a result of each and every timing or logic error captured at the gate-level.

On the other hand, we are at the threshold of an explosion in new data, produced not only by large, powerful scientific and commercial computers but also by the billions of low-power devices of various kinds (AGRAWAL et al., 2016). Notably, the video content is ruling this data explosion due to its intrinsic data-intensive characteristic. Globally, the total video traffic will represent 80% of all internet traffic by 2022, up from 70% in 2017 according Cisco (2018). The video compression integration into the Systems on Chip (SoCs) is becoming further required every year. On the other hand, video compression applications are becoming more complex due its continuously improving. High Efficient Video Coding (HEVC) video coding standard can double the compression efficiency of the previous H.264/AVC (Advanced Video Coding) standard while increasing the encoder

[1] In this work, the term closed-loop refers to the feedback loops of algorithms with dynamic behavior and their hardware accelerators, analog to the closed-loop of linear dynamic systems & signals.

complexity by approximately a factor of three compared with its predecessor H.264/AVC (GRELLERT et al., 2013).

A video decoder must be *standardized* to guarantee the compatibility of the video bitstream format for any device compliant with the respective standard. On the other hand, to effectively search the video redundancies, a video encoder aggregates much more complexity than a decoder. The video encoder VLSI (Very Large Scale Integration) design space trading-off hardware performance by the compression efficiency (i.e., quality versus compression ratio curve) has been a challenger research issue leveraged by the many compute-intensive algorithms, decisions, and coding tools (MARTINA, 2019). Internally, the encoder must also implement decoding responsible for closing the loop by reconstructing the reference frames used in the motion search engine. The decoding loop must strictly seek standardization to preserve the full compliance of the output video stream.

Video encoding is a great challenge for AxC and TS hardware design. The effects of the errors into the joint hardware accelerator algorithm are in closed-loop and must be bridged during the encoder application's runtime. For a broad class of algorithms, as in video encoding, errors in the current state of the hardware accelerators are propagated, affecting the decision and actions of subsequent processing steps into the software, potentially adversely affecting the application quality. Therefore, the closed loop between the hardware accelerator and the algorithms requires online error feedback during the application runtime to accurately quantify the impact of the induced errors.

The central objective of this thesis is to investigate AxC and TS hardware design techniques on accelerators and their impacts in a real-world application case. Therefore, this thesis presents a novel framework capable of crossing both logic and timing errors induced by AxC and TS between the design layers up to the algorithms and application. In other words, the framework connects the accelerator at the gate-level and the application software dynamically. Thus, enabling a realistic evaluation of the complex iterations between the joint algorithm-accelerator and the overall application. By going beyond state of the art, the framework proposal offers a holistic approach aiming at maximizing the energy savings or performance improvement while still fulfilling the good-enough quality of any application (i.e., also in cases involving accelerator-algorithm closed-loops).

1.2 Thesis Contributions

This work presents the following novel contributions:

- On the approximate Discrete Tchebichef Transform (DTT) hardware architectures (PAIM et al., 2019):
    - A power-, area-, and compression-efficient approximate DTT hardware design using pruning and truncation keeping the same compression-efficiency levels.
    - An efficient pruned transposition buffers tradeoff exploration which, in turn, guarantees, for all proposed approximations, both less power dissipation and smaller circuit area with improvements in terms of the maximum frequency.
    - A discussion about the tradeoff between circuit area and power dissipation versus quality-compression in the prunes of the design space exploration of the proposed DTT.
- This thesis shows a new cross-layer framework from gate-level to the application-level, supporting the evaluation of both AxC and TS hardware design (i.e., under logic- and timing-relaxation). To achieve that, our implementation bridges the gap between the timed gate-level simulations and the application software dynamically, correctly crossing these layers to support the investigation of all kinds of (a) timing-speculative hardware design, (b) approximate computing, and (c) dynamic algorithms behavior (i.e., including algorithm-accelerator closed-loops).

  Case study 1 – Approximate computing design: On the Approximate Adders for trading logic errors by higher energy-efficiency with lower circuit area, shown in (PAIM et al., 2020).

    - This case study investigates the power dissipation vs. coding efficiency tradeoff analysis with more than 3,000 approximate SAD architectures, considering the impact of circuit-derived approximation errors while simulating the entire video coding process.
    - We explore both truncation and copy approximate adder variants in the SAD architecture, to examine the tradeoff between coding efficiency vs. power dissipation.

Case study 2 – Timing-speculative design: Trading temperature- and aging-induced errors by higher performance (PAIM et al., 2021c).

- This case study investigates how timing-speculative design and its timing errors impact the runtime of the joint algorithm-accelerator operating in a closed-loop.
- We investigate three existing BMA of the x265 encoder IME implementation and their joint operation with a SAD accelerator. We conclude that – under the same degradation-induced timing errors – both diamond search (DS) and hexagon search (HS) are most reliable than the uneven multi-hexagon search (UMHS).

Case study 3 – Timing-speculative design: Trading VOS-induced timing errors by higher energy-efficiency (PAIM et al., 2021a).

- This case study investigates how the VOS-induced timing errors impact at runtime the joint algorithm-accelerator operating in a closed-loop. This novelty opens new investigations about VOS-induced timing errors embracing any application. In the case study of video coding herein investigated, this is work preserves full compliance to the standard - a mandatory requirement in this application.
- We investigate the VOS-induced timing errors in a SAD hardware accelerator running into a video encoder. We observe the ultimate impact in the coding efficiency loss and conclude that the design requires adopting a VOS more conservative than state of the art. We also find that the minimum timing guardbands (TGBs) necessary to sustain a tolerable loss depends on the motion quantity. This work demonstrates that the greater is the video motion quantity, the wider must be the TGB to fulfill a good enough quality of service.

1.3 Outline

The remaining of this thesis is organized as follows: The approximation-tolerant video processing application under evaluation is presented in **Chapter 2**. **Chapter 3** presents an overview about power dissipation parameters, approximate computing techniques, and timing-speculative hardware design. **Chapter 4** presents efficient approximate DTT architectures for ASIC (Application Specific Integrated Circuit) and FPGA (Field Programmable Gate Array) hardware design. **Chapter 5** presents a cross-layer framework for exploiting approximate computing and timing-speculative hardware design. In this same chapter, results regarding application quality versus speed/energy-efficiency gains on the use of (a) approximate adders, (b) temperature and aging-induced timing errors, (c) voltage over-scaling are shown. Finally, conclusions and future works are drawn in **Chapter 6**.

2 VIDEO COMPRESSION OVERVIEW

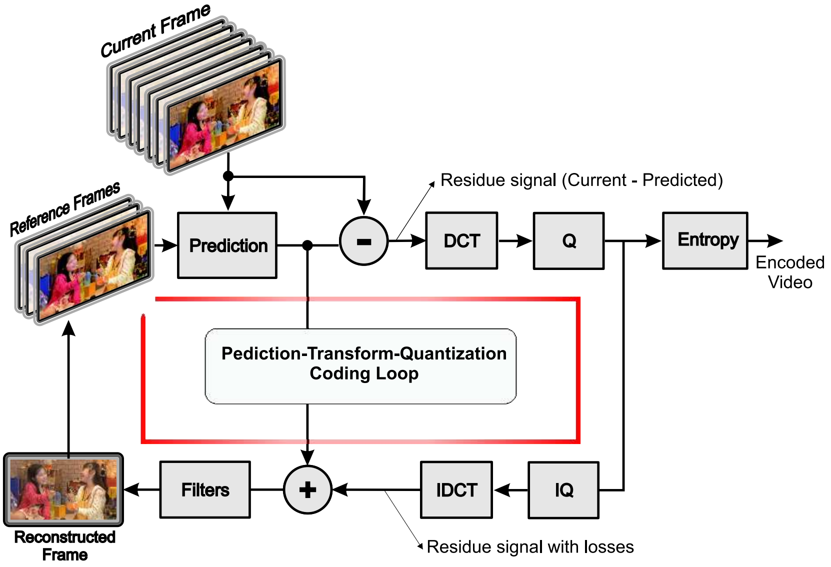
Error-tolerant applications, such as multimedia and video coding, can process the information with lower-than-standard accuracy at the circuit level while still fulfilling a good and acceptable service quality at the application level. This thesis adopts a video processing application as a case study to investigate approximate computing and timing speculation hardware design. Therefore, this chapter presents a brief overview of video compression, focusing on the Integer Motion Estimation (IME) and Discrete Cosine Transform (DCT).

High Efficiency Video Coding (ITU-T; ISO/IEC, 2013) is a current video coding standard, which is capable of doubling the compression efficiency of the previous H.264/AVC (Advanced Video Coding) standard (ITU-T; ISO/IEC, 2011) for the same quality (SULLIVAN et al., 2012), while ensuring that the increase in the encoder complexity does not exceed that of the H.264/AVC by a factor of three (GRELLERT et al., 2013). The higher compression efficiency of HEVC comes from improved partitioning structures and more sophisticated algorithms (more prediction modes, discrete transform sizes, optimized entropy coding, etc.).

HEVC is a hybrid video compression scheme that combines block-based spatial and temporal prediction of frames with transform coding of the prediction residue and an entropy encoder (NGUYEN et al., 2013). An important innovation in the HEVC standard is the generic, quadtree-based approach to divide a picture for prediction and transform coding, which is described in the following (NGUYEN et al., 2013). Figure 2.1 illustrates the whole encoding loop in the current state-of-the-art hybrid encoders as HEVC.

Such innovations' main drawback is that they require several computations to be performed, which directly translates into high power dissipation and energy consumption. This becomes a serious issue when battery-powered devices that support high-definition videos, e.g., smartphones, tablets, smart-watches, camcorders, are considered. For these platforms, and energy-efficient hardware design is quintessential to make better use of battery resources. Thus, developing hardware accelerators for video codec modules is of utmost importance to reduce the power/energy requirements of HEVC video codecs.

Figure 2.1: Video coding loop.


Current Frame
Reference Frames
Prediction
Residue signal (Current - Predicted)
DCT
Q
Entropy
Encoded Video
Pediction-Transform-Quantization Coding Loop
Reconstructed Frame
Filters
IDCT
IQ
Residue signal with losses

Source: The Author.

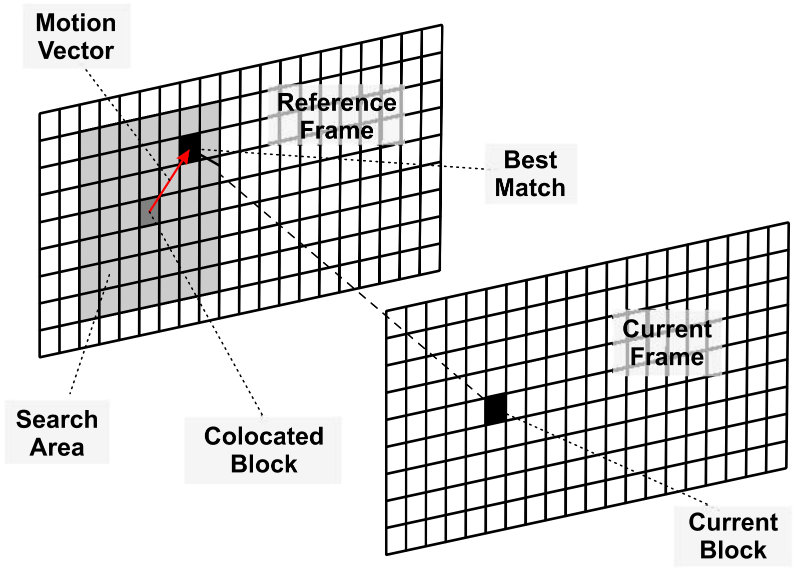
2.1 Integer Motion Estimation

The HEVC encoding splits a frame into several coding tree units (CTUs). Then, a mode decision is applied to each CTU. A CTU can be partitioned into several coding units (CUs), whose dimensions typically range from 64×64 to 8×8 blocks. A second partitioning decision occurs in each CU during prediction, forming the prediction units (PUs) (WANG et al., 2017). Motion estimation is applied at the PU level. This stage is responsible for deciding the initial center of the search, which will be used for the next step. The algorithm chooses among many candidates, including vectors related to the temporal and local (spatial) neighboring PUs. The SAD is computed for every block associated with these candidate vectors, and the smallest one will have its vector defined as the initial center of the search.

Eight PU types are supported in HEVC: four symmetric motion partitions (SMPs) and four asymmetric motion partitions (AMPs). Fig. 2.3 depicts all the partitioning structures, from CTU to PU level, showing the sizes of each PU for a 16×16 CU.

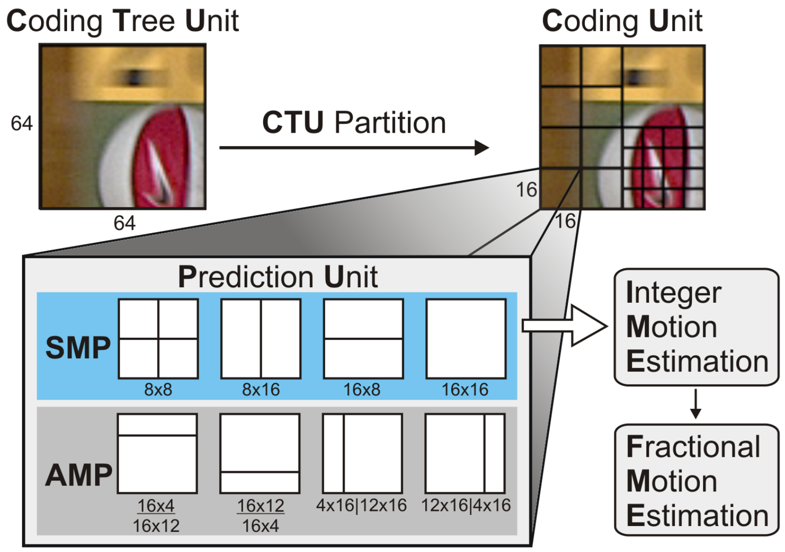
The number of PU blocks for a 64×64 CTU is detailed in Table 2.1. Every possibility adds up to 593 block partitions for a single CTU, which translates to 2372 ME calls if four reference frames are used. This number explains the reason why SAD execution time in software is predominant compared to other operations. Note that fast HEVC encoders use heuristics that skip some of these possibilities, but doing that does not completely eliminate this issue.

Figure 2.2: Motion estimation example between two frames.



Source: Adapted from Porto (2008a).

Figure 2.3: Coding tree unit and the partitioning types.



Source: The Author.

Table 2.1: PU amount of each size in a 64×64 CTU

Partition	#PUs	Partition	#PUs	Partition	# PUs
8 × 4	128	**16 × 16**	16	**32 × 64**	2
8 × 8	64	**16 × 32**	8	**64 × 32**	2
8 × 16	32	**32 × 16**	8	**48 × 64 ***	2
16 × 8	32	**24 × 32 ***	8	**64 × 48 ***	2
12 × 16 *	32	**32 × 24 ***	8	**16 × 64 ***	2
16 × 12 *	32	**8 × 32 ***	8	**64 × 16 ***	2
4 × 16 *	32	**32 × 8 ***	8	**64 × 64**	1

* Asymmetric motion partitions (AMP)

IME employs a search algorithm, usually within a delimited region of the reference frame (i.e., the search area), to find the most similar block using SAD as the criterion. The most trivial algorithm is called full search (or exhaustive search), which matches every possible block in the search area. Several fast search algorithms were designed to speed up this process, such as diamond search and hexagon search.

(a) Diamond Search (DS): The algorithm searches for four candidates around the search center. The center of the search is updated to the best candidate among the tested ones (see Fig. 2.4(a)). The execution stops when none of the four candidates evaluated are better than the current center or when the search reaches the search area's border.

(b) Hexagon search (HS): this BMA is split into two sequential stages, which are illustrated in Fig. 2.4(b):

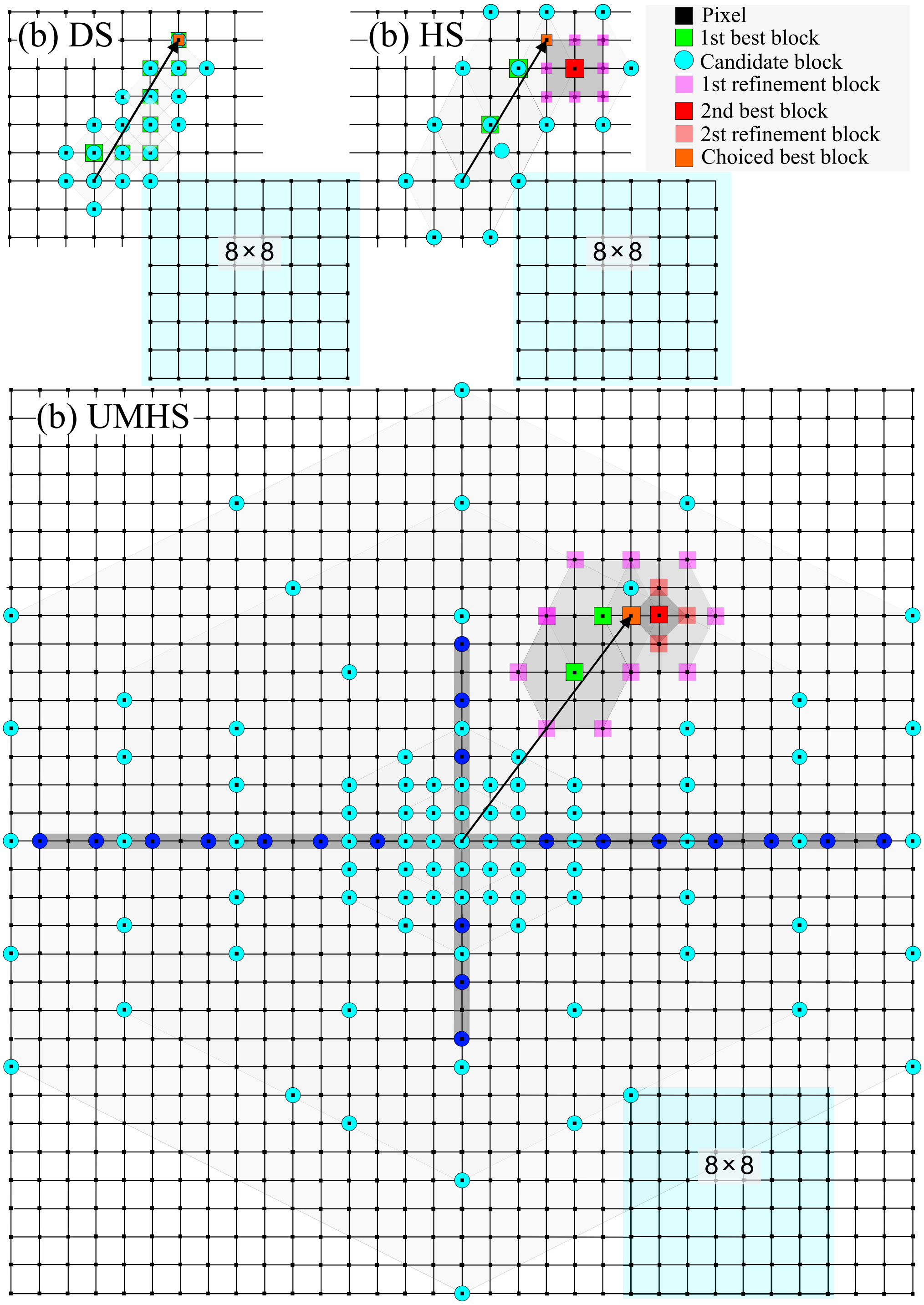
(Step 1) Hexagon: performs a six-point search in a hexagon-shaped format around the center of the search. The center is updated whenever a candidate is more similar to the block being encoded than the one in the current center. This stage stops when none of the candidates are better than the current center. Since the second iteration of this stage, only three candidates are evaluated after three of the points will always have been evaluated previously.

(Step 2) Square Refinement: An eight-point square refinement is applied around the current center. The final vector value is defined by the best candidate evaluated at this stage.

(c) Uneven multi-hexagon search (UMHS): this algorithm has a more complex flow than DS and HS. The UMHS can be split into three main stages, as follows:

(Step 1) Small diamond: this stage performs up to three diamond-shaped iterations (with four candidates each), the first one in the initial MV and two others in the co-localized vector and the best current MV, in case they have not been tested. The whole algorithm may early-terminate if the best SAD value is already below a pre-defined threshold, in which case it performs a cross-search – Fig. 2.4(c) – and another diamond iteration before halting. Otherwise, the BMA adapts the search range based on the best SAD value and the current best MV. The first diamond is executed around the best current MV from the SVI. Two conditional diamond iterations are performed in both the co-localized vector and the best current MV once again if none of them are the same as the initial MV so that no redundant searches are performed. Then, the algorithm attempts to early-terminate its execution whenever the smallest SAD value (the one from the best current candidate) is already below some pre-defined thresholds. In such cases, the algorithm may perform a cross-search, whose shape is shown in Fig. 2.4(c) as well as an additional eight-point diamond-shaped iteration before ending its execution. Suppose the execution does not terminate due to the SAD thresholds. In that case, the BMA performs an adaptation in the size of the search range based on the variability of both the SAD magnitude and the vector coordinates of neighboring MVs.

Figure 2.4: BMA examples: (a) Diamond search (DS) (b) Hexagon search (HS) and square refinement; (c) Uneven multi-hexagon search (UMHS).



Source: The Author.

(Step 2) Cross diamond: this second stage performs a cross-search and a four-point square search around the current best vector.

(Step 3) Multi-hexagon: this last stage performs a sixteen-point hexagon-shaped search, as presented in Fig. 2.4(c) until the BMA hits the boundaries of the search range. If the best vector found is outside the original search range, the BMA executes the entire HS BMA described earlier.

2.1.1 Sum of Absolute Differences

Several metrics can be applied to determine the degree of similarity between the two blocks. They differ in ease of implementation, efficiency, and accuracy, i.e., how precisely they can define the blocks' similarity.

Sum of Absolute Differences (SAD) is the most straightforward metric used in the video encoding process. It is applied by calculating the differences between the co-localized pixels of a current and a candidate block, performing a whole operation in these differences, and then adding the values. SAD is employed in the IME and is also one of the most used metrics in the video encoding process, representing, on average, 22.4% of the encoding time in the HM reference software (ABREU et al., 2017). The complete equation is given by 2.1, in which O and R denote the current and the candidate blocks, respectively; m and n refer to the width and height of the blocks (which have the shape of the PU being considered).

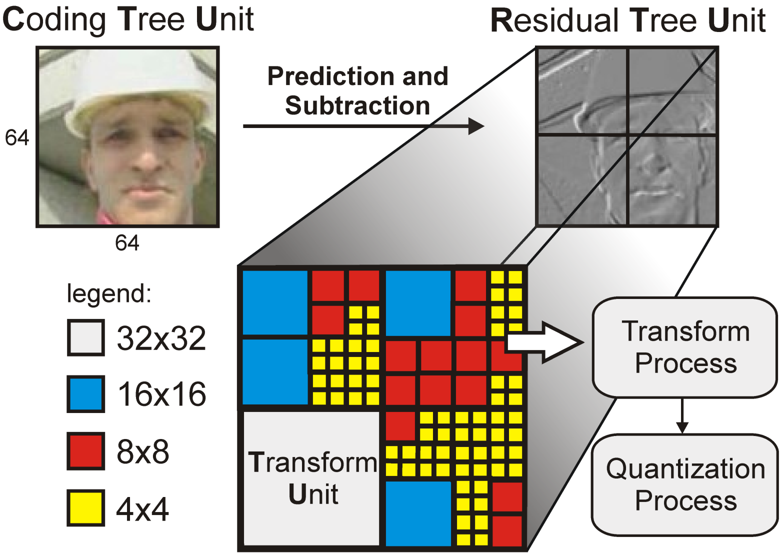
The equation for SAD calculation is given in (2.1). O is the original block (the block in the current frame). R is the reference block (from a set of many candidate blocks in the reference frames), m and n are the block's dimensions in samples. The details of the SAD hardware architecture of (2.1) are given in section IV.A.

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |O_{i,j} - R_{i,j}| \tag{2.1}$$

2.2 Discrete Cosine Transform

HEVC encoder process split each frame into equal-sized blocks, called Coding Tree Units (CTUs). The CTUs are further divided into Coding Units (CUs), which can have different partitioning sizes. During the prediction (Fig. 2), a second partitioning

Figure 2.5: Coding tree unit and the TU partitioning types.



Source: The Author.

decision occurs in each CU, forming the prediction units (PUs) (WANG et al., 2017). The current block is subtracted from the predicted results in the residual.

The DCT stage in the HEVC standard is performed based on the Residual Quad-Tree (RQT). For each of these different CU partitioning sizes, a set of DCTs of different sizes have to be executed. Upon the decision of the best CU partitioning size, the loop is performed to produce the reconstructed frame, which will be used as a reference for the following current frame to be predicted.

To summarize, the DCT processes the information coming from the residue created between the prediction stages – which manages to find spatial/temporal redundancies – and the original video. It is executed several times to find the best combination to be sent to the output.

Figure 2.5 presents the scheme of the relation between the tools of modern hybrid video encoders, such as the H.264/AVC and HEVC standards, in a more detailed manner. The most costly stage regarding computational effort and execution time in the state-of-the-art HEVC standard is the inter-frame prediction. The inter-frame prediction generates a residual block, which is the difference between the original and reference blocks. This residual block is then used by the encoder to recover the encoded block, to be used as a reference for the next frame's inter-frame prediction. Therefore, an encoder-decoder dependency loop is created inside the encoder.

The DCT is responsible for processing the information from the residual block created in these previous prediction stages by its execution in different sizes of Transform

Units (TUs) in the RQT. HEVC supports squared-shaped TU sizes of 4×4, 8×8, 16×16, and 32×32. In the context of modern video encoders, the transform stage can represent a bottleneck for the encoder. It is part of the encoder-decoder loop, so its forward and inverse transform have to be performed in each step of the encoding process – both to be sent to the output and decoded used by the inter-prediction stage. Since the DCT can represent a critical step in the video encoding process, it is crucial to develop dedicated hardware modules for this stage.

The DCT model developed for standards such as H.264/AVC and HEVC is focused on maintaining the orthogonality by using integer coefficients, allowing for a more straightforward hardware design without multipliers (BUDAGAVI et al., 2013) and avoiding mismatching between coders and decoders, which is called coding drift errors. Unlike H.264/AVC, HEVC applies a higher amplification in the integer coefficients to reduce each coefficient's quantization error and increase the compression efficiency, shortening the gap from the ideal floating-point implementation.

The 2-D Discrete Cosine Transform can be based on the separability property to process the blocks line by line. This separated approach uses two 1-D DCT stages with a transposition buffer that separates both stages. At every cycle, one line of the input block is read from the frame/residue and feeds the first-stage DCT. The transposition buffer enables both stages of the 2-D DCT to operate independently, allowing a higher level of parallelism. While the first 1-D transform operates on the input block and the processed data is stored in the transposition buffer, the second 1-D transform operates on the transposition buffer's outputs.

2.3 Encoder Quality of Service Metrics

Distortion metrics, or comparison criteria, are used to quantify the quality of the encoded video. This is a complex parameter to be defined and evaluated, as it can be both subjective and objective (SALOMON; MOTTA, 2010). Subjective methods of comparison are based on human judgment and are operated without an explicit criterion (SALOMON; MOTTA, 2010). The objective criteria are based on comparing the pixels of the original image with the pixels of the image after encoding (i.e., reconstructed image). This comparison is made frame by frame (SALOMON; MOTTA, 2010). The criterion most used and accepted by the scientific community is the PSNR (Peak Signal-to-Noise Ratio) (SALOMON; MOTTA, 2010), defined by Equation 2.2. In Equation 2.2, MAX is the maximum value that a luminance sample can reach ($2^n - 1$, where n is the number of bits to represent each sample). MSE is the mean quadratic error (*Mean-Squared Error*), defined in Equation 2.3, where *M* and *N* are the dimensions of the image in pixels and *O* and *R* are the original and reconstructed images, respectively (SALOMON; MOTTA, 2010).

$$PSNR_{dB} = 20 * \log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right) \tag{2.2}$$

$$MSE = \frac{1}{M.N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(R_{i,j} - O_{i,j})^2 \tag{2.3}$$

The adequate metrics to evaluate the quality of service of an encoder is to measure its coding efficiency that associates quality and compression in the same metric. The HEVC standard analysis is usually run with QPs 22, 27, 32, and 37, as stated by the common test conditions (CTCs) (BOSSEN et al., 2013) recommendation. The runs with these four QPs generate the four points required to generate two coding efficiency measurement metrics specific for video compression technology: Bjontegaard-delta bitrate (BD-BR) and Bjontegaard-delta peak signal-to-noise ratio (BD-PSNR) (BJONTEGAARD, 2001). BD-BR indicates how higher/lower the bitrate should be to achieve the same PSNR quality. Positive values indicate an increase in the bitrate, which is not desired. On the other hand, BD-PSNR corresponds to the average PSNR difference, in dB, for the same bitrate. Negative values of BD-PSNR mean less quality for the same bitrate, which is not desired. Details about the BD-BR and BD-PSNR functions can be found for Excel in ITU-T (2001) (i.e., original from ITU-T meetings) and Matlab in Serge (2013).

2.4 Conclusion

This chapter presented an overview of the video encoding principles and modules. The last two chapters of this thesis investigate hardware architectures dedicated to specific video encoder processing kernels (i.e., DCT and SAD architectures). The concepts presented in this chapter provide a background for understanding the impacts of applying AxC and TS hardware design techniques in the video encoder.

3 HARDWARE DESIGN CONCEPTS AND TECHNIQUES BACKGROUND

This chapter presents the background of hardware design concepts and techniques. The first section revises the power dissipation in CMOS circuits. Then, the second section presents a review of approximate computing techniques. The third session shows a review of timing-speculative hardware design. Finally, the last section concludes this chapter.

3.1 Power dissipation in CMOS circuits

Power dissipation is one of the major concerns when designing digital circuits. This section discusses the most influential parameters of power dissipation in CMOS circuits. There are three primary sources of power dissipation in CMOS circuits under normal operation (WESTE; ESHRAGHIAN, 1994): (1) power dissipation caused by leakage currents and sub-threshold currents, (2) short-circuit power dissipation, which occurs due to the direct current flow from the power supply to the ground during the switching process in a transistor gate, and (3) dynamic power dissipation, which is the result of loading and unloading the capacitances of the circuits.

Even in sub-micron processes, the significant component of total power is the dynamic power dissipation considering a normal operation[1]. Therefore, this section focus on dynamic power dissipation. Dynamic power dissipation is the result of charging and discharging the capacitances of the circuit due to the switching activity, which is intrinsic to the operation of CMOS circuits (see 3.1). Therefore, dynamic power dissipation is the major component in high-performance circuits (i.e., circuits operating at high clock frequency) as in video processing. This part of the chapter will describe a realistic power evaluation methodology using a commercial digital flow applied to any digital circuit. As a case study, this methodology will be illustrated within the context of a multimedia application by considering actual video sequences as input stimuli. The main aspects that contribute to power dissipation in CMOS circuits are presented below.

Assuming a logic gate G connected to a load capacitance C_L, the gate output value changing will cause a current flow that will either charge or discharge C_L. This cycle will dissipate power according to Equation 3.1, where A is the output node activity measured

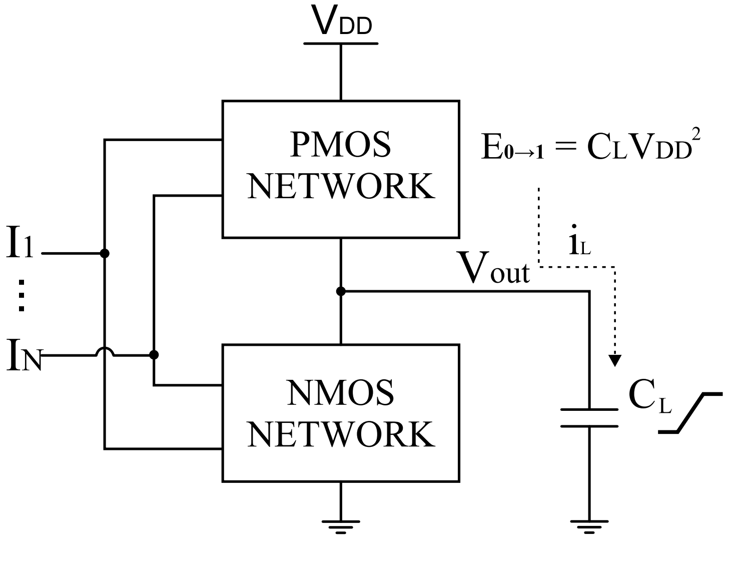
[1] Operating at high clock frequencies and nominal voltage.

in events/second for a complete charge/discharge, and f_{clk} is the clock frequency.

$$P_{Total} = P_{Static} + \frac{1}{2}C_L A {V_{dd}}^2 = P_{Static} + \frac{1}{2}\alpha f_{clk} C_L {V_{DD}}^2 \tag{3.1}$$

For a digital circuit operating at frequency f, most of the circuit nodes will not change at every clock cycle. Conversely, it is most likely that these nodes will change their values with a probability α, which represents the node activity factor. Considering a circuit with more than one node, the value of α is computed as a function of input statistics and logical models that describe the device. The energy drained from the power supply for a $0 \rightarrow V_{DD}$ transition at the output of a CMOS gate is given by $C_L {V_{DD}}^2$. Figure 3.1 shows a circuit model highlighting the switching power component (CHANDRAKASAN; BRODERSEN, 1995).

Figure 3.1: Dynamic power characterization of a CMOS gate.



Source: Adapted from Chandrakasan and Brodersen (1995).

It is noticeable that approximately half of the energy drained from the power supply is stored in the load capacitor. The other half is dissipated in the PMOS network as heat. For a $V_{DD} \rightarrow 0$ transition at the output, there is no charge drained from the power supply, and the energy stored in the capacitor $(C_L V_{DD}^2/2)$ is dissipated in the NMOS pull-down network (CHANDRAKASAN; BRODERSEN, 1995).

If a simple transition is performed at each clock cycle at the rate f_{clk}, then power is given by $(C_L V_{DD}^2 f_{clk}/2)$. However, there are cases in which the signal transition occurs at different frequency rates. We have to consider the value of the number of transitions per clock cycle or node transition activity factor. In this case, Equation 10 represents the average power, which corresponds to the average number of switching transitions in a period, where $\alpha_{0\to 1} = \alpha_{1\to 0} = \alpha$. Since the internal nodes of a circuit may switch their values, the transition activity has to be calculated for all nodes of the circuit, and the total power of the circuit is given according to Equation 3.2.

$$P_{Total} = P_{Static} + \sum_{i=1}^{\#\ nodes} \frac{1}{2}\alpha_i C_i {V_{DD}}^2 f_{clk} \tag{3.2}$$

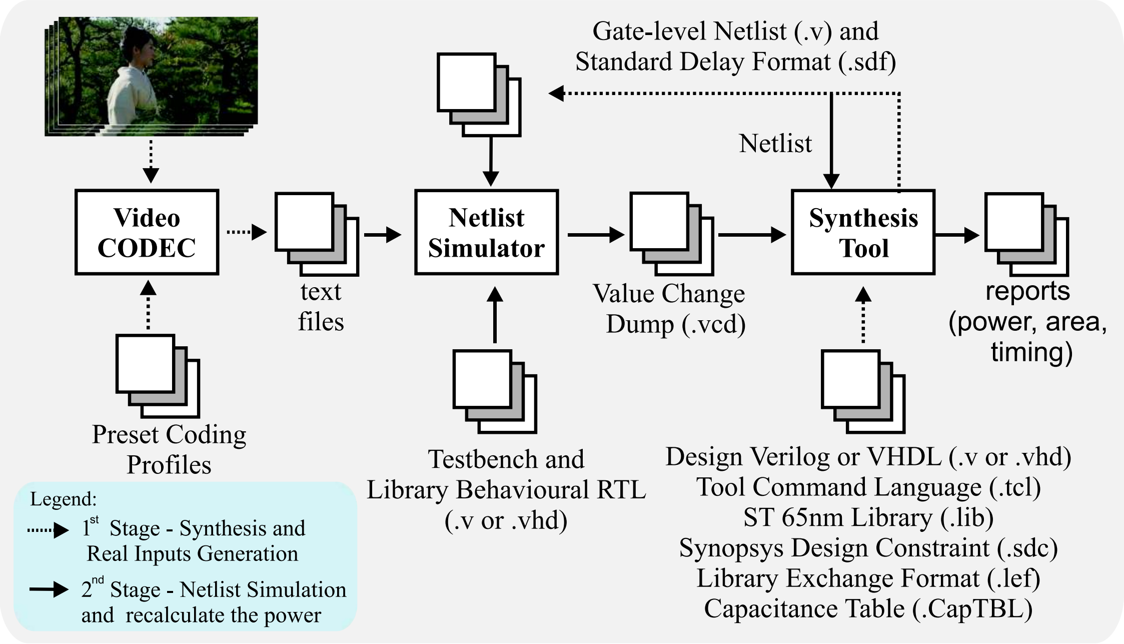
Signal transitions cause dynamic power dissipation in digital integrated circuits. Commercial synthesis tools generate, by default, probabilistic input values to estimate the power dissipated by the circuit. This approach, however, is often pessimistic and rarely represents the actual input behavior of the application for which the circuit was conceived. Therefore, an accurate power estimation methodology is necessary to represent the actual application behavior and provide a realistic quality of critical power-hungry blocks.

Figure 3.2 shows the current industrial methodology for power dissipation extraction. First, the Register-Transfer Level (RTL) description is synthesized using a commercial synthesis tool – such as the Cadence Genus Synthesis Solution$^{\text{TM}}$ and Synopsys Design Compiler$^{\text{TM}}$ – for a given technology. As a by-product, the tool generates the circuit netlist in Verilog, the Standard Delay Format (SDF) file that annotates the specific delays for gates and nets and the design reports of cell area, power dissipation, and Critical Path Delay (CPD). Then, the simulation tool – like Cadence Incisive$^{\text{TM}}$ and Synopsys VCS – simulates the generated netlist with the testbench files – written in VHDL/Verilog/SystemVerilog – feeding the circuit with input data from standard benchmark images or videos to generate the dump file that is in either VCD (Value Change Dump) or TCF (Toggle Count Format) files.

The input data used as stimuli are extracted from the target application software and stored in text files. With these text files, the simulation tool receives these values and executes the testbench with the gate-level netlist files and SDF delay files to generate accurate switching activity. At the end of the simulation, the dump file is generated containing all nodes' switching activity within the circuit netlist. Lastly, the synthesis tool is executed a second time with the same parameters. In this turn, the dump file is fed to the software to generate accurate power dissipation reports.

Modern synthesis tools support the estimation of the gate interconnection penalties in the circuit area, power dissipation, and delay. Cadence Genus logic synthesis tool disposes of the physically-aware logic synthesis through the physical layout estimator (PLE) mode, while the Synopsys Design Compiler tool offers the Topographical mode

Figure 3.2: Current industrial ASIC Design flow for data-driven power estimation.


Gate-level Netlist (.v) and
Standard Delay Format (.sdf)
Netlist
Video CODEC
text files
Netlist Simulator
Value Change Dump (.vcd)
Synthesis Tool
reports (power, area, timing)
Preset Coding Profiles
Testbench and Library Behavioural RTL (.v or .vhd)
Design Verilog or VHDL (.v or .vhd)
Tool Command Language (.tcl)
ST 65nm Library (.lib)
Synopsys Design Constraint (.sdc)
Library Exchange Format (.lef)
Capacitance Table (.CapTBL)
Legend:
1st Stage - Synthesis and Real Inputs Generation
2nd Stage - Netlist Simulation and recalculate the power

Source: The Author.

to analyze the impact of interconnections. These tools estimate the length of the nets and take into account the load capacitance effects in the power dissipation, considering a relatively pessimistic layout routing estimation. Such an analysis requires the inclusion of the Library Exchange Format (LEF) files, mainly containing the library's physical layout information. LEF macro includes the inner library cell's capacitance, and the tech LEF comprises the process metal capacitance for the interconnection capacitance estimation. Additionally, standard-cell libraries usually offer an additional file, namely capacitance table, which describes the technology capacitances more precisely and fine-grained. It considers the process variations.

FPGA vendors as Xilinx and Altera also integrate a data-driven power estimation option in their design flow. FPGA industrial flows to estimate power is very similar to ASIC. It allows the circuit's delay consideration (i.e., informing SDF) and the interconnection estimation inclusion for the targeted device. In this thesis, we also employ data-driven power estimation when evaluating the results for FPGA.

3.2 Approximate Computing

The approximate computing paradigm emerged as a key alternative for trading off accuracy and energy efficiency. Error-tolerant applications, such as multimedia and signal processing, can process the information with lower-than-standard accuracy at the circuit level while still fulfilling a good and acceptable service quality at the application level.

In a conventional digital circuit design, one usually assumes that the system will provide accurate results, even under fixed-bit-width arithmetic operations. However, in practice, operations with a high level of accuracy are not always necessary. Even the analog-to-digital conversion at the start already computes approximate data w.r.t. the physical phenomenon. In many applications, digital circuits can generate good enough results, instead of the most accurate results, without compromising the functioning of the application in the system as a whole (ZHU; GOH; YEO, 2009). The approximate computing paradigm emerges as a solution to exchange precision, or quality of results, to reduce power dissipation, energy consumption, and area in the VLSI circuit.

The most inherently adopted approximation in practice for significant savings in power is reducing bit-widths by truncation. This strategy proposes reducing the bit width of architecture to obtain a more energy-efficient circuit without reducing the information quality. In general, we can apply this approach to some arithmetic blocks in the circuit, where the resulting values may be less than expected. We can also use this strategy in the entire circuit to design it to operate with a certain width that works appropriately for a specific application, even with the introduction of errors in the processed results' magnitude.

Another strategy related to approximate computing, which has shown significant results in the design of low-power digital circuits, is pruning. According to (SCHLACHTER et al., 2017), circuit pruning is a design technique that consists of removing logic circuit blocks and their associated wires. The strategy aims to establish a relationship between the computation accuracy, the power dissipated, the area, and the circuit's delay. The amount of pruning to be applied depends on the target application's error tolerance. According to (ZHANG et al., 2018b), for a computing unit, the pruning of a node reduces the switching activities of the circuit. It omits some of the cells that produce the node logic, resulting in reduced energy consumption. If the node is in the CPD of the circuit, there will naturally increase operating speed. In this case, the price to be paid is the increased inaccuracy of the results at the circuit level, which must be evaluated in the application.

3.2.1 Approximate Adders

Adder circuits are key logic blocks in hardware accelerators in many error-tolerant applications (GUPTA et al., 2013; SOARES et al., 2019), so the use of approximate adders (AAs) is a very appealing alternative for hardware implementations. Notably, the use of approximate arithmetic computing in the video accelerators appears as a promising solution for increasing energy efficiency while keeping high-level perceptual information.

The approximations in adders lead to circuit simplifications, which may reduce both CPD and power dissipation. There are two types of AAs: those focused on improving computational performance (i.e., faster circuits) by breaking the carry chain, and those focused on reducing energy consumption by simplifying the Less Significant Bits (LSBs). The former usually features more straightforward approximate computation, presenting highly frequent small-magnitude error characteristics, according to to (HUANG; LACH; ROBINS, 2012). Conversely, the latter is based on multiple blocks of operands, which can lead to the occurrence of higher magnitude errors. In either case, the correct parameterization of the approximate block bit-length can control the error magnitude. In the following subsections, we describe all the AA explored in this work.

Truncation AA: the truncation technique is the most straightforward approximation since the LSBs are pruned or statically set to 0-logic. The remaining MSBs (Most Significant Bits) are implemented as an accurate and conventional adder topology. This adder is named TRUNC$_0$ (Fig. 3.3a). In this work, we have also tested the LSBs set statically to one (TRUNC$_1$ – Fig. 3.3b).

Copy AA: in (GUPTA et al., 2013), the authors proposed a topology where the result of the approximate directly copy one part of one input operands and the carry-in estimation for the precise part of the adder copy the most significant bit. Fig. 3.3c illustrates this topology where the most significant bit of the operand A (highlighted in red) is copied to the carry-in of the precise part. This strategy achieves a 75% chance of correct carry-in estimation (GUPTA et al., 2013). In this work, we named this adder as the copy adder following the convention adopted in (SOARES; COSTA; BAMPI, 2016).

Copy A$_{\mathbf{Copy}}$ (Fig. 3.3c): The approximate part is a copy of k LSBs of operand A, and its MSB is set as the carry-in for the precise part. The B$_{Copy}$ adder (Fig. 3.3d) follows the same approach, but it uses the operand B as the reference.

Copy A$_{\mathbf{AND}}$ (Fig. 3.3e): The approximate part is a copy of k LSBs of operand A. The carry-in estimation for the precise part is defined as the AND operation between the

MSBs of the approximate part of both input operands. The copy B$_{AND}$ adder (Fig. 3.3f) follows the same approach, but it uses operand B as the reference.

Copy A-B$_{\mathbf{AND}}$ (Fig. 3.3g): The result of the approximate part comes from the alternate copy of operands A and B bits from the LSB to the MSB, starting with operand A. The carry-in estimation for the precise part is defined as the AND operation between the MSBs of the approximate part of both input operands. The copy B-A$_{AND}$ adder (Fig. 6h) follows the same approach, but it starts copying the LSB from operand B.

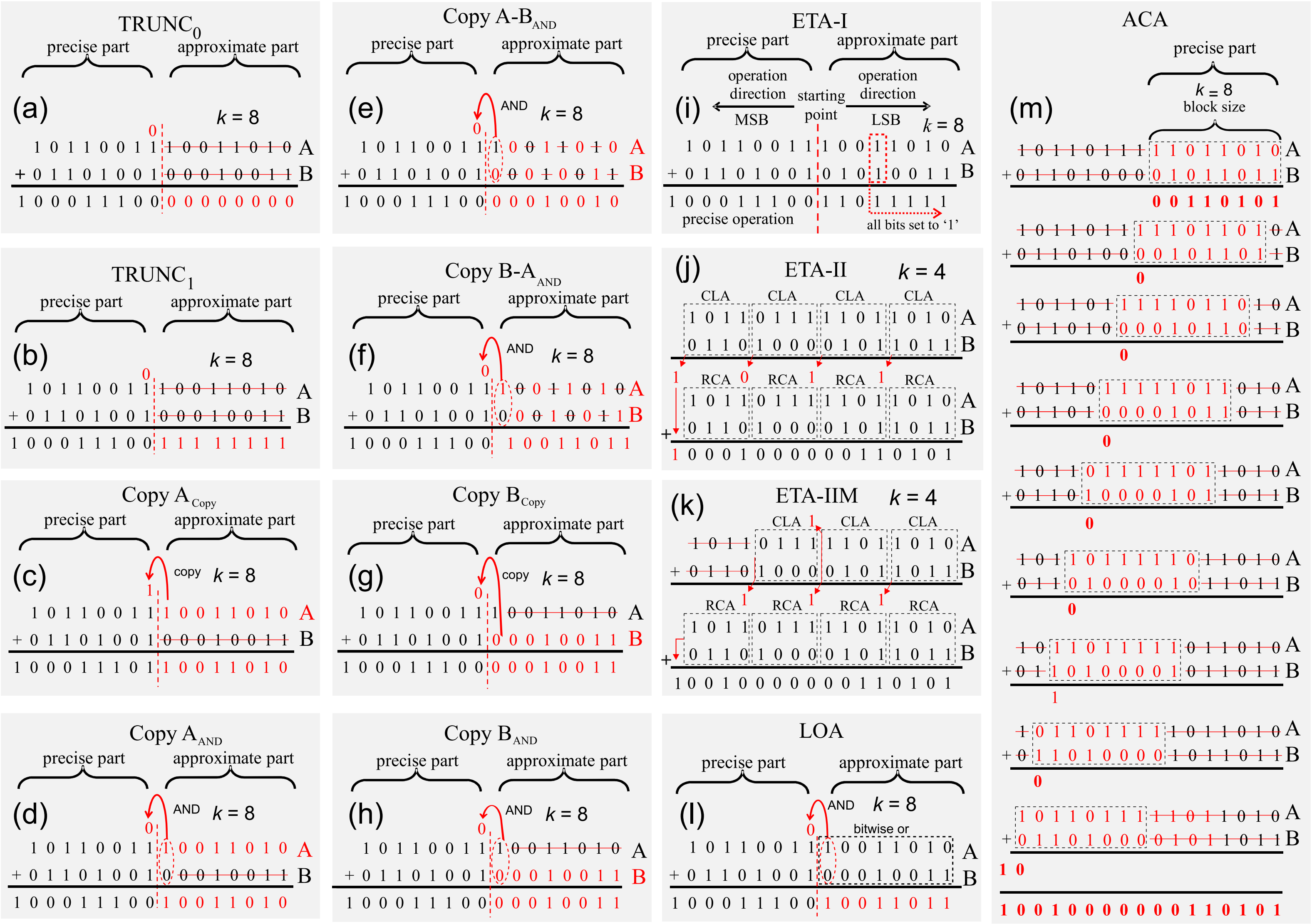
LOA AA: In this scheme, OR gates are used rather than full-adder cells to compute the result of the approximate part (MAHDIANI et al., 2010). The carry-in estimation for the precise part is computed through the carry-generate operation between the $(k-1)$-th position of both input operands. This technique grants a 75% probability of estimating the carry-in correctly. Fig. 3.3i shows an example of the LOA addition scheme.

ETA-I AA: This scheme was proposed in (ZHU et al., 2010) and it employs two adders: (a) a regular carry-propagating adder for the precise part, and (b) a carry-less sum on the approximate part, which uses only half adders. In the approximate part, the half adders compute the sum from the MSB to the LSB. When the first carry-generate operation is equal to 1-logic, all the remaining least significant sum bits are set to 1-logic, as presented in Fig. 3.3j. In the precise block, the operation computes the sum without a carry-in to simplify the circuit.

ETA-II AA: Non-overlapping independent blocks compose the ETA-II (ZHU; GOH; YEO, 2009). According to Fig. 3.3k, this adder is divided into M blocks of K bits. The method to attenuate error in sum result is based on the use of K-bit carry speculation scheme, which uses the carry look-ahead adder (CLA). The CLA block adopts the carry-in value in 0-logic, and the carry-out value is used to speculates the carry-in of K bits to its next most significant K-bit ripple-carry adder (RCA) block. The K-bit RCA block generates the addition result. This addition scheme allows reducing the CPD that is composed of K-bit CLA and K-bit RCA. The work in (ZHU; GOH; YEO, 2009) also proposes a modified structure of the ETA-II adder, called ETA-IIM (Fig. 3.3l) to improve the accuracy of ETA-II. ETA-IIM AA includes an improved MSBs calculation, reducing the probability of the higher magnitude errors caused by the non-overlapping condition present on its predecessor ETA-II. In the modified design, the most significant parts, i.e., the more upper order bits, are calculated more accurately than the lower ones. This means that the length of carry-in speculation for the most significant parts can be larger than the least significant ones.

ACA AA: The ACA (VERMA; BRISK; IENNE, 2008), uses K-bit overlapping blocks to divide the AA. The concept of the block in ACA was described according to its implementation, where K is the size of the adder block necessary to generate the current bit based on the speculation of the K-1 last bits. In the first block, there is the computation of the first eight LSBs sum result (K = 8 in Fig. 3.3m). Alternatively, the next blocks (from 1 up to N-K) compute only a 1-bit sum result. Therefore, there is the computation of each 1-bit sum result in ACA architecture by considering carry chain speculation of K-1 bits.

Figure 3.3: The set of AAs examples used in SAD to evaluate the impact on the HEVC video encoding.



Source: The Author.

3.2.2 Transform Coefficient Approximations

The work presented in (BAYER; CINTRA, 2010) introduced a new algorithm, called Rounded Cosine Transform (RCT), that tries to eliminate the computational complexity of the DCT. This new algorithm approximates the cosine function rounding each DCT multiplication matrix term to the closer integer. Considering the closer integer x of the real number, we can consider the function $[cos(x)]$ as an approximation to the cosine. Note that the $[cos(x)]$ simplify the DCT matrix in values 0, 1 and -1 which the multipliers can be implemented by simple additions (BAYER; CINTRA, 2010). The RCT takes advantage of those trivial operations by reducing the computational effort of the operations in hardware. The resultant matrix (3.3) of the RCT coefficients is presented in (BAYER; CINTRA, 2010).

According to (3.3), the RCT presents lower computational complexity than DCT, since only 0, 1, and -1 values are presented in the matrix. Its implementation can be easily done by using only adders/subtractors and one right bit-shifting for the division by 2. In (BAYER; CINTRA, 2010), it is shown that even with the losses of rounding, the RCT presents comparable and even better results in some cases compared with the DCT. Therefore, the RCT performance combined with the lower computational complexity shows that this transform represents an excellent choice for dedicated hardware for image compressing.

The work in (CINTRA; BAYER, 2011) improve the RCT orthogonality.

$$\mathbf{RCT} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix} \tag{3.3}$$

An improved RCT implementation was presented in (POTLURI et al., 2014) employing only 14 additions and named Modified Rounded DCT (MRDCT). The adders counting reduction is given by a matrix with zeros, presented in (3.4):

$$\mathbf{MRDCT} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.4}$$

DTT can be seen as an orthogonal transformation from the discrete Tchebichef polynomials (DTP) (BATEMAN et al., 1953). In fact, the DTT presents a polynomial kernel and maps a finite sequence of data onto the DTP space (ISHWAR; MEHER; SWAMY, 2008). The main properties of the DTP, such as orthogonality, recurrence, and normalization, have been pointed in (PRATTIPATI et al., 2013; PRATTIPATI; SWAMY; MEHER, 2013). By these properties, a kernel function generates a matrix – similarly to the DCT generation – proposed by (PRATTIPATI et al., 2013; PRATTIPATI; SWAMY; MEHER, 2013). The kernel function is used in the DTP domain, where a transformed sequence of an input data sequence is reached. The complete evaluation of the DTT is more detailed in (PRATTIPATI et al., 2013; PRATTIPATI; SWAMY; MEHER, 2013). The resultant matrix for an 8-point is given in (3.5), where $\mathbf{F}_8 = diag(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{42}}, \frac{1}{\sqrt{42}}, \frac{1}{\sqrt{66}}, \frac{1}{\sqrt{142}}, \frac{1}{\sqrt{546}}, \frac{1}{\sqrt{66}}, \frac{1}{\sqrt{858}})$ (PRATTIPATI et al., 2013; PRATTIPATI; SWAMY; MEHER, 2013).

$$\mathbf{P} = \frac{1}{2}\mathbf{F}_8 \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -7 & -5 & -3 & -1 & 1 & 3 & 5 & 7 \\ 7 & 1 & -3 & -5 & -5 & -3 & 1 & 7 \\ -7 & 5 & 7 & 3 & -3 & -7 & -5 & 7 \\ 7 & -13 & -3 & 9 & 9 & -3 & -13 & 7 \\ -7 & 23 & -17 & -15 & 15 & 17 & -23 & 7 \\ 1 & -5 & 9 & -5 & -5 & 9 & -5 & 1 \\ 1 & 7 & -21 & 35 & -35 & 21 & -7 & 1 \end{bmatrix} \tag{3.5}$$

A pruned DTT was proposed in (KOUADRIA et al., 2017), and it consists in removing the last four lines of the exact DTT presented in (3.5), which results in (3.6).

$$\mathbf{P}_{Pruned} = \frac{1}{2}\mathbf{F}_8 \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -7 & -5 & -3 & -1 & 1 & 3 & 5 & 7 \\ 7 & 1 & -3 & -5 & -5 & -3 & 1 & 7 \\ -7 & 5 & 7 & 3 & -3 & -7 & -5 & 7 \end{bmatrix} \tag{3.6}$$

The CB-2015 forward DTT matrix is non-orthogonal. Hence its inverse is different from the forward transpose transform. The bottleneck of the CB-2015 approximation is the inverse matrix that presents all coefficients with non-zero values coefficients. As ± 3 values are used in the matrix; therefore more additions are required in its hardware implementation. Since the CB-2017 DTT matrix is a quasi-orthogonal approximation, its inverse matrix can be approximated as the forward transpose. The CB-2017 forward and inverse matrices are composed of ± 2, ± 1, and 0 values.

Based on the simplified approaches of the approximate DTTs with a reduced number of adders and their intrinsic lower computational efforts, the approximate DTT-based algorithms of (OLIVEIRA et al., 2015) and (OLIVEIRA et al., 2017) are excellent choices for image compressing hardware design.

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.7}$$

The CB-2017 DTT matrix, in (3.8), is a quasi-orthogonal approximation, thus its inverse matrix is approximated as the forward transpose with a correction factor after the diagonal matrix $(\mathbf{T} \cdot \mathbf{B} \cdot \mathbf{T}^{\top}) \cdot \mathbf{D}$, where $\mathbf{D} = (1/4) * diag(\frac{1}{8}, \frac{1}{3}, \frac{1}{3}, \frac{1}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{3}, \frac{2}{5})$ and $\mathbf{B}$ is the input pixels block. The diagonal correction is processed in other coding steps, such as the quantization block.

CB-2017 proposes an iterative search to choose efficient coefficients that generate a parametric class of possible DTT matrices. Firstly, CB-2017 applies a modification in the exact DTT – with real coefficients whose equations are described in (SENAPATI;

MAHAPATRA, 2014) – by multiplying the diagonal matrix **D** to reduce the high dynamic range of the DTT coefficients. The parametric transform class was generated varying an α factor with a real range from 0 to $\frac{5}{2}$ in the equation $round(\alpha \cdot \mathbf{D} \cdot \mathbf{T})$, where $\mathbf{T}$ is the exact DTT with real coefficients. The resulting matrix was generated with $\alpha = 2$, relaxing the orthogonality and observing that the diagonal deviation results in rounded integer coefficients in the range of $0, \pm1, \pm2$.

$$\mathbf{T} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ -2 & -1 & -1 & 0 & 0 & 1 & 1 & 2 \\ 2 & 0 & -1 & -1 & -1 & -1 & 0 & 2 \\ -2 & 1 & 2 & 1 & -1 & -2 & -1 & 2 \\ 1 & -2 & 0 & 1 & 1 & 0 & -2 & 1 \\ -1 & 2 & -1 & -1 & 1 & 1 & -2 & 1 \\ 0 & -1 & 2 & -1 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -2 & 1 & 0 & 0 \end{bmatrix} \tag{3.8}$$

To the best of our knowledge, there is currently no available exploration of the fractional power-of-two coefficients for better approximations and for improving the hardware design efficiency – such as bit-width reduction – as shown in this work. Table 3.1 shows a summary of related works results, listing the compression efficiency (quality-compression) metrics and corresponding ASIC and FPGA implementation results. Both CB-2015 (OLIVEIRA et al., 2015) and CB-2017 (OLIVEIRA et al., 2017) do not perform an entropy analysis in order to evaluate the compression. This thesis shows an analysis that includes the quality-entropy evaluation of the proposed transforms and compares them to state-of-the-art proposals. Both Exact DTT (PRATTIPATI et al., 2013; PRATTIPATI; SWAMY; MEHER, 2013) and K-2017 (KOUADRIA et al., 2017) present only results for GPP, thus lacking ASIC and FPGA implementation results. CB-2015 (OLIVEIRA et al., 2015) presents only FPGA results. CB-2017 (OLIVEIRA et al., 2017) presents a naive power extraction methodology that neither considers realistic input vectors nor it takes into account the wire and internal gate delays, which translates to very inaccurate power estimation values.

Table 3.1: Summary of Related Work Results.

	Quality and Compression-efficiency					**ASIC**			**FPGA**			**Design**
Related Work	PSNR	SSIM	SR-SIM	Quality-Entropy	Subjective Qual.	Power	Timing	Area	Power	Timing	Area	Pruned Buffers
$\mathbf{DTT}_{\mathbf{Exact}}$	✓	×	×	×	×	×	×	×	×	×	×	×
$\mathbf{DTT}_{\mathbf{CB\text{-}2015}}$	×	✓	✓	×	✓	×	×	×	✓	✓	✓	×
$\mathbf{DTT}_{\mathbf{CB\text{-}2017}}$	×	✓	✓	×	✓	✓	✓	✓	×	✓	✓	×
$\mathbf{DTT}_{\mathbf{K\text{-}2017}}$	✓	✓	×	×	✓	×	×	×	×	×	×	×
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

3.3 Timing-Speculative Hardware Design

Digital circuit designers usually protect their circuits against timing errors imposing design-time guardbands to conventional employing a positive time slack value (i.e., zero minimal) with a worst-case static timing analysis (STA) for the variability conditions (i.e., environmental, reliability and process). Fig. 3.4-(a) shows the conventional design of exact circuits imposing the worst-case STA at the design time.

Timing-speculative (TS) hardware design allows increasing the frequency or decreasing the voltage beyond the limits determined by STA, thereby removing pessimistic safety margins that conventional design implement to prevent timing errors (ASSARE; GUPTA, 2019).

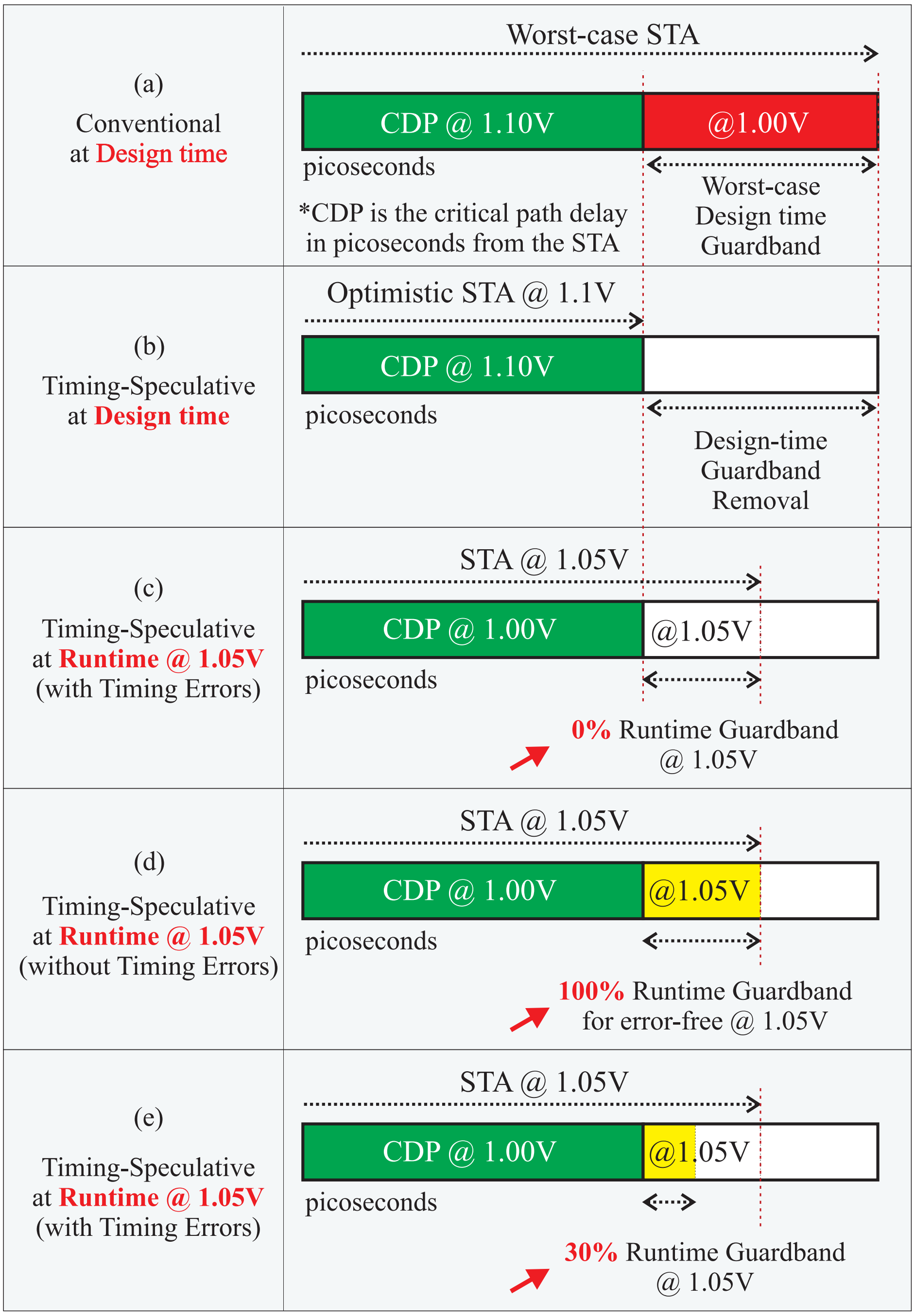
Unlike the traditional design, TS hardware design narrows the design-time guardbands for environmental and variability conditions, which generate circuits more optimist than the worst-case scenario. Hence, this consideration produces unprotected runtime conditions in which the circuit produces timing errors.

Fig. 3.4-(b-e) shows an example to explain the timing guardbands (applied at design time or runtime) in the TS hardware design when approaching voltage over-scaling. Fig. 3.4-(b) shows the TS hardware design considering and optimistic STA for 1.10V removing the guardband protection for 1.00V at design time. Fig. 3.4-(c) illustrates the TS hardware design at runtime with the circuit operating at 1.05V, without protections applied at runtime. On the other hand, the timing errors caused by the design-time considerations can become unacceptable to the application in specific cases. In these cases, runtime guardbands can be applied by decreasing the operational clock frequency with a penalty of energy efficiency. Fig. 3.4-(d) demonstrates the runtime guardbands employing to recover the error-free operation at 1.05V.

In this work, we consider the case of 100% runtime guardband the slack time value (in picoseconds) added at runtime on top of the designed to embrace all timing paths for the specific runtime variability conditions. In Fig. 3.4-(d) example, the runtime guardband added for error-free operation @1.05V is the slack time which comprehends the difference between the design-time (STA @ 1.10V) and the current runtime (STA @ 1.05V). An example of narrowing the runtime guardband to 30% for an operation at 1.05V is shown in Fig. 3.4-(e). The narrowed runtime guardband (i.e., narrower than 100%) can be manipulated at runtime depending on the operational conditions for recovering the accuracy required by the application.

Notice that these timing errors prediction still considers the worst-case process variability. In average cases, the timing errors in practice tend to be less aggressive than predicted due to the process variability distribution. The following subsections present the timing effects of transistor aging, temperature rising, and voltage over-scaling, explored in the last chapter of this thesis.

Figure 3.4: Example of the timing guardbands for voltage over-scaling at design time for the (a) conventional hardware design, and (b) TS hardware design. Example of runtime guardbands to the TS hardware design (c-e).



Source: The Author.

3.3.1 Aging and Temperature Effects

The delay increase caused by temperature rising on transistors is an unavoidable problem intrinsic to semiconductor devices operation. Further, another significant collateral effect of temperature rising is the sharp increase of transistor susceptibility to physical degradation, i.e., aging effects, during the hardware accelerator lifetime (KEANE; KIM, 2011). Aging effects lead to an increase in voltage threshold (V_T) and a reduction in carriers mobility (μ), both of which contribute to a growth in the hardware accelerator delay. Therefore, nanoscale hardware accelerators operating on timing edges can lead to wrong output results if device degradation is not considered (EBRAHIMI et al., 2013).

The impact on performance caused by the increase in temperature during runtime must be thoroughly analyzed when designing hardware accelerators since high temperatures increase the device delay, which results in timing errors on hardware accelerators due to the unsustainable clock frequency (CHOUDHURY et al., 2014). The introduction of timing guardbands mitigates these effects, i.e., an additional timing slack included on top of the maximum delay target of the hardware accelerator (EBRAHIMI et al., 2013; AMROUCH et al., 2019), ensuring that the hardware accelerator reliability, as the temperature-induced delays increase, will still fall within the guardband margin. Despite its proven success, this approach has a considerable drawback that restrains the hardware accelerator from running at its maximum performance. Nonetheless, many compute-intensive and error-resilient applications like machine learning, digital signal processing, and multimedia systems (HE; GERSTLAUER; ORSHANSKY, 2013) can cope with added noise without any loss of functionality at the expense of result quality degradation (STANLEY-MARBELL et al., 2020). These applications allow a tradeoff exploration where the performance is boosted with the guardbands removal, introducing some quality loss due to eventual timing errors induced by temperature rising.

Accurate gate-level simulation is mandatory to analyze the effects of degradations (e.g., temperature and aging) on the delay of hardware accelerators. This process requires degradation-aware cell libraries that model such effects (AMROUCH et al., 2019). The cell libraries employed in this work for considering aging and temperature effects are based on 14nm FinFET (Fin Field-Effect Transistor) technology. The underlying transistor devices were fully calibrated to match 14nm FinFET measurements from Intel technology process.

14nm FinFET degradation-aware cell libraries: These libraries are based on standard EDA tool flows. They are characterized for multiple case scenarios of temperature and device wear. The delay information of every standard cell (combinational and sequential gates) is obtained in the scope of studied degradation.

This work targets a 14nm FinFET technology, where both p-FinFET and n-FinFET transistor devices were calibrated with Intel 14nm measurement data using semiconductor physics-based technology CAD (TCAD) simulations. The calibration employs the industry-standard compact model (BSIM-CMG) (DUARTE et al., 2015) to match the library measurements with the Intel measurement data.

Details on the transistor modeling and calibration are available in (MISHRA et al., 2019). The cells were created employing the schematics of the cell libraries in the Silvaco open-source PDK (SILVACO, 2019) set to use our calibrated FinFET models. The library characterization was based on accurate SPICE simulations to consider all the electrical properties of the transistors. This approach also enables the exploration of the BTI (Bias Temperature Instability) effect on transistor aging since it is a major concern in current and future technologies (MISHRA et al., 2019). In this work, we consider a device lifetime of 10 years to estimate aging effects.

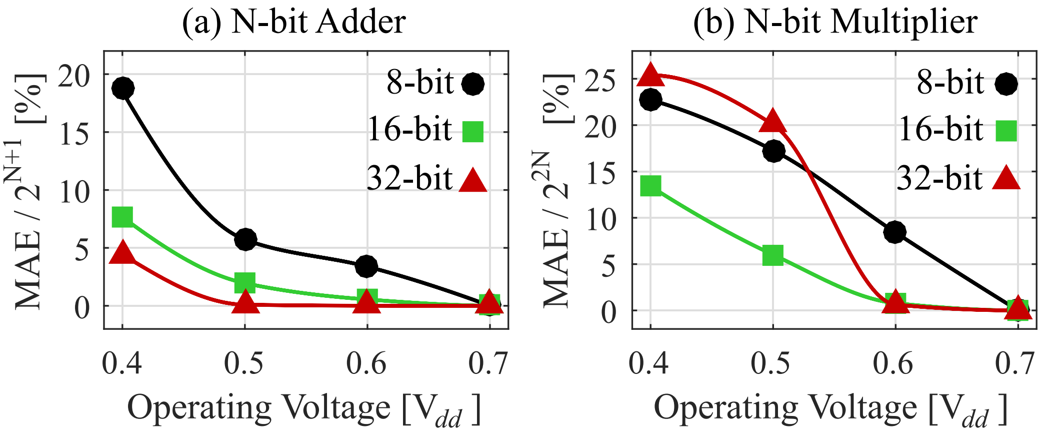
Every standard cell has been characterized for 7×7 input signal slews and output load capacitance according to typical industry standards adopted by major PDK vendors. Our set of degradation-aware cell libraries includes stand-alone temperature models as well as composite models that embed the impact of both temperature and aging altogether. All generated libraries are fully compatible with existing commercial EDA tool flows (e.g., Synopsys and Cadence). Therefore, designers can directly employ them to perform accurate timing analysis of their hardware accelerators and perform an accurate GLS.

3.3.2 Voltage Over-Scaling

Voltage Over-Scaling (VOS) is a widely used TS design technique since it can be seamlessly applied to any circuit, and it delivers high power savings. VOS is applied by keeping the operating clock frequency constant and decreasing the voltage supply below its nominal value (ZERVAKIS et al., 2018).

The power dissipation of a circuit depends quadratically on the voltage value, as shown in (3.2), and thus VOS can deliver high power reduction. On the other hand, as the voltage value decreases, the circuit becomes slower, and errors are generated due to the circuit paths that fail to meet the timing requirement. For example, in Fig. 3.5, we plot the error generated due to VOS for N-bit adders and N-bit multipliers for N equal to 8, 16, and 32. The examined circuits, in Fig. 3.5, are synthesized at their highest achievable clock frequency (i.e., zero lack) targeting the 7nm FinFET technology library (CLARK et al., 2016). The operators are automatically selected by the synthesis tool targeting performance. To evaluate the error of the examined circuits when applying VOS, we employ the mean absolute error (MAE) metric that is calculated as (3.9) shows.

Figure 3.5: Errors generated due to VOS in (a) N-bit adder and (b) N-bit multiplier arithmetic circuits with respect to the voltage value.



Source: The Author.

Since the examined circuits feature varying bitwidths, we plot in Fig. 3.5, the Normalized MAE. Normalized MAE is obtained by normalizing MAE with respect to the output bitwidth of each circuit, i.e., divide MAE by 2^{N+1} for the adders and by 2^{2N} for the multipliers. The nominal voltage value is 0.7V, and we examine the VOS application with a 100mV step. As shown in Fig. 3.5, the error due to VOS is small for small voltage decreases (almost negligible in most cases) due to the small number of paths that violate the timing constraint. However, after a voltage level, the number of violating paths increases significantly, leading to large error values. For example, the error of the 32-bit adder is almost zero at 0.5V, i.e., 200mV decrease, but becomes 5% at 0.4V. Similarly, the error of the 32-bit multiplier is zero at 0.6V and becomes 20% at 0.5V. As illustrated in Fig. 3.5, after a voltage level, the induced error increases so fast that it impedes the exploitation of the full spectrum of power reduction that the VOS application could potentially deliver.

$$MAE = \frac{1}{n}\sum_{i=0}^{n-1} |Exact_i - Approx_i| \tag{3.9}$$

3.4 Conclusion

This chapter presented a background on the sources of power dissipation in digital circuits, approximate techniques, and timing-speculative hardware design techniques. The first section demonstrated the key design parameters that influence the power dissipation and show data-driven methods of current industrial flows to improve the power estimation. The second section is shown approximate computing concepts and a review about approximate adders hardware design. The last section presented timing-speculative hardware design concepts with a review of temperature- and aging-induced effects and the voltage over-scaling technique. Concluding, this chapter reviewed hardware design concepts that are fundamental to the presentation of the following chapters. The following two chapters contain the novelties and the key contributions of this thesis.

4 APPROXIMATE PRUNED AND TRUNCATED DISCRETE TCHEBICHEF TRANSFORM HARDWARE DESIGN

Recently, the Discrete Tchebichef Transform (DTT) has emerged as a lower complexity discrete transform for picture coding, with characteristics close to the DCT (NAKAGAKI; MUKUNDAN, 2007; ISHWAR; MEHER; SWAMY, 2008; RAHMALAN; ABU; WONG, 2010; KHONGSIT; RANGABABU, 2017). This transform presents a polynomial kernel whose characteristics such as high energy compaction and decorrelation make it comparable to the DCT, especially when specific image features that profoundly influence the reconstructed image quality – like its structure and content – are taken into account (RAHMALAN; ABU; WONG, 2010). The exact integer 8-point DTT was first proposed in (PRATTIPATI et al., 2013).

Several works proposed approximated 8-point DTT matrices for picture coding to reduce computing complexity to obtain power and time savings. In (OLIVEIRA et al., 2015), named CB-2015, (OLIVEIRA et al., 2017), named CB-2017, and (KOUADRIA et al., 2017), named K-2017 in this thesis, the prime purpose is to reduce the number of arithmetic operations while keeping the information quality. The matrices are simplified by restricting the coefficient values, so the multiplierless hardware implementations are simplified and can be implemented using only word-shifts, adders, and subtractors, all operating in parallel. This approach then enables dissipation savings in the hardwired DTT when encoding an image since it presents a lower computational effort than the native DCT. K-2017 (KOUADRIA et al., 2017) proposes a DTT approximation by pruning the last four lines from the original transform matrix (PRATTIPATI et al., 2013; PRATTIPATI; SWAMY; MEHER, 2013) which reduces the amount of hardware.

Given the state-of-the-art approximate 8-point DTT described in the matrix (3.8), all approximations proposals derive from such matrix. Considering operations with integers, the smaller magnitude of coefficients causes truncation in the internal transform calculations and leads to lower values for the non-diagonal residues, which reduces non-orthogonality. Our first approach consists of dividing the entire matrix by power-of-two values – since they translate to efficient shift operations in hardware – to reduce the datapath bitwidth.

The power-of-two values chosen were selected, considering three objective quality metrics - PSNR, Y-SSIM, and SR-SIM - while testing fifty images from the same repository used in CB-2017 (OLIVEIRA et al., 2017). The quality metrics kept stable as we proceed from $\frac{1}{2}$ up to $\frac{1}{16}$, and when using $\frac{1}{16}$ the quality was higher than that of (OLIVEIRA et al., 2017). Using $\frac{1}{32}$ caused significant degradation in quality. Therefore, all arithmetic operators of the transposition buffer and at the second 1-D transform stage are reduced by, at least, four bits when using the $\frac{1}{16}$ coefficient, which significantly reduces area and power and can potentially improve the system throughput. Matrix (4.1), named Proposal 0, uses this approach.

$$\mathbf{T}_{p0} = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ -\frac{1}{8} & -\frac{1}{16} & -\frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{1}{16} & \frac{1}{8} \\ \frac{1}{8} & 0 & -\frac{1}{16} & -\frac{1}{16} & -\frac{1}{16} & -\frac{1}{16} & 0 & \frac{1}{8} \\ -\frac{1}{8} & \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} & \frac{1}{16} \\ \frac{1}{16} & -\frac{1}{8} & 0 & \frac{1}{16} & \frac{1}{16} & 0 & -\frac{1}{8} & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{8} & -\frac{1}{16} & -\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & -\frac{1}{8} & \frac{1}{8} \\ 0 & -\frac{1}{16} & \frac{1}{8} & -\frac{1}{16} & -\frac{1}{16} & \frac{1}{8} & -\frac{1}{16} & 0 \\ 0 & 0 & -\frac{1}{16} & \frac{1}{8} & -\frac{1}{8} & \frac{1}{16} & 0 & 0 \end{bmatrix} \tag{4.1}$$

Dividing the matrix by sixteen requires a final multiplication by two hundred fifty-six in the original $\mathbf{D}$ diagonal correction matrix, resulting in $\mathbf{D}_0 = \mathbf{D} * 256$. Finally, our correction is $(\mathbf{T}_{p0} \cdot \mathbf{B} \cdot \mathbf{T}_{p0}^{\top}) {\cdot} \mathbf{D}_0$ with $\mathbf{D}_0 = 64 * diag(\frac{1}{8}, \frac{1}{3}, \frac{1}{3}, \frac{1}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{3}, \frac{2}{5})$.

Another way to simplify the datapath is to reduce the number of operations it shall perform to reach the computation result. Thus, this work proposes three other versions of (4.1) by removing rows from the reference matrix. Lines six, seven and eight, contain coefficients correlated to high-frequency components that have less significance to the human visual system and tend to be eliminated by quantization. The removal of these lines enables a considerable hardware reduction in the transform implementation. Hence, this work explores the effects of removing either one, two, or three of such matrix rows.

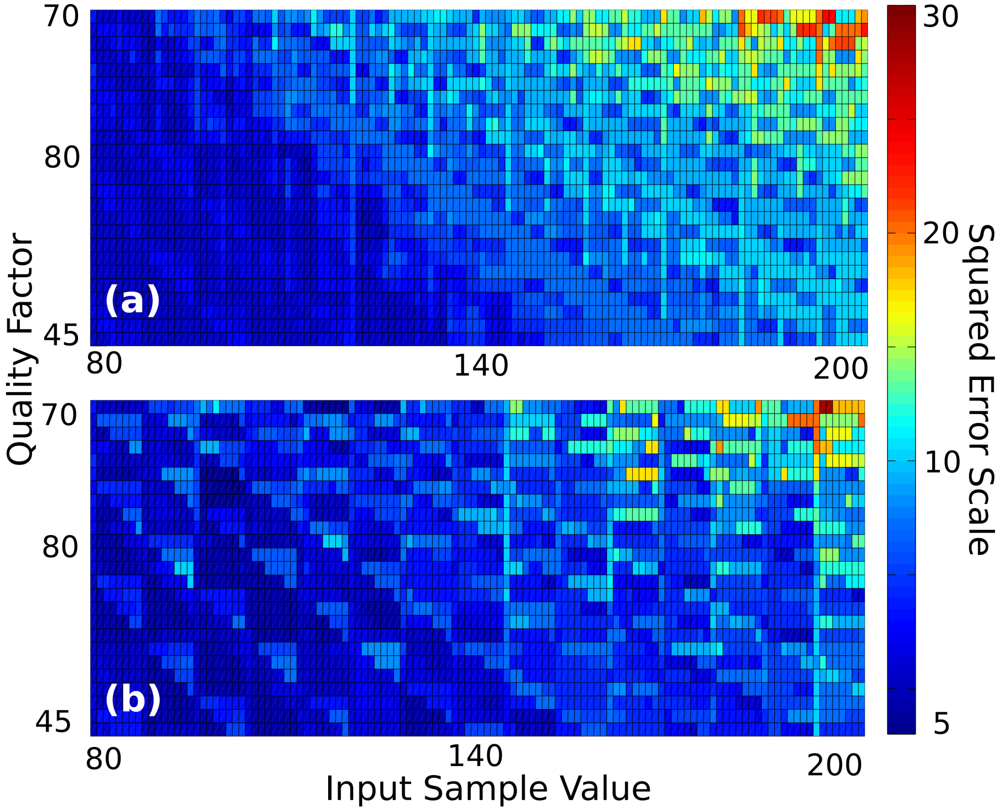
Matrix 4.2 shows a 3-line pruned version of the reference matrix which now has the diagonal correction $(\mathbf{T}_{p3} \cdot \mathbf{B} \cdot \mathbf{T}_{p3}^{\top}) {\cdot} \mathbf{D}_3$, where $\mathbf{D}_3$ is represented by $\mathbf{D}_3 = 64 * diag(\frac{1}{8}, \frac{1}{3}, \frac{1}{3}, \frac{1}{5}, \frac{1}{3})$.

$$\mathbf{T}_{p3} = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ -\frac{1}{8} & -\frac{1}{16} & -\frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{1}{16} & \frac{1}{8} \\ \frac{1}{8} & 0 & -\frac{1}{16} & -\frac{1}{16} & -\frac{1}{16} & -\frac{1}{16} & 0 & \frac{1}{8} \\ -\frac{1}{8} & \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} & \frac{1}{16} \\ \frac{1}{16} & -\frac{1}{8} & 0 & \frac{1}{16} & \frac{1}{16} & 0 & -\frac{1}{8} & \frac{1}{16} \end{bmatrix} \tag{4.2}$$

4.1 A Metric for Non-orthogonality Evaluation

CB-2017 (OLIVEIRA et al., 2017) presented a metric to evaluate the approximate 8-point DTT non-orthogonality. However, the analytic equation fails since it does not take into account the truncation effect. The truncation is different for each one amplitude of the input samples. The evaluation of the squared diagonal $\mathbf{T}_{p0} \cdot \mathbf{T}_{p0}^{\top}$ deviation from the identity matrix $\mathbf{I}$ was done according to (4.3). Figure 4.1 evaluates (4.3) for both quantization factor (Q_F) and input sample amplitude (k).

Figure 4.1: Non-orthogonal Squared Deviation Distribution per Q_F and Input Amplitude for: (a) CB-2017; (b) Proposal 0.



Source: The Author.

Integer-based calculations truncate the fractional values of the matrix coefficients, resulting in lower non-diagonal residue values that reduce the non-orthogonality. Figure 4.1 depicts the distribution of diagonal deviation residue of the discrete transform. The diagonal deviation residue is more distributed in the proposed transform – named Proposal 0 – compared with CB-2017 since more distributed blue regions are found in Figure 4.1. This means that the Proposal 0 transform is more orthogonal for most cases.

$$\mathbf{S}_{\mathbf{Error}} = \sum [\frac{\mathbf{T}_{p1} \cdot (k\mathbf{I}) \cdot \mathbf{T}_{p1}^{\top}}{\mathbf{Q}} - \frac{(k\mathbf{I})}{\mathbf{Q}}]^2 \tag{4.3}$$

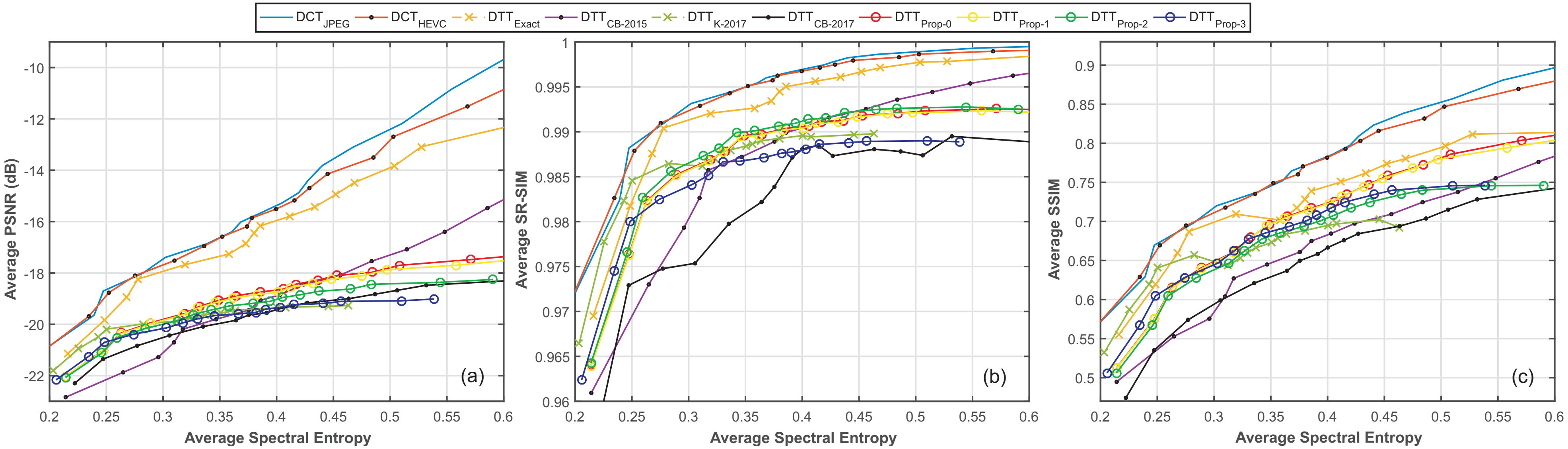
4.1.1 Objective Compression Efficiency Evaluation

JPEG standard defines that each pixel block is divided element-wise by the quantization matrix $\mathbf{Q}$ to obtain the quantized coefficients (OLIVEIRA et al., 2017). This quantization matrix is given by $\mathbf{Q} = \lfloor (S \cdot \mathbf{Q_0} + 50)/100 \rfloor$ where Q_0 is the base quantization table. The S factor defines the modification on the base table, and its value depends on the Q_F value. If Q_F is less than 50, we have $S = 5000/Q_F$, otherwise $S = 200 - 2 \cdot Q_F$. The Q_F factor directly impacts image quality and image compression ratio. As the Q_F value increases, image quality is improved, but the compression ratio is penalized. This factor ranges from 0 (lowest image loss, highest compression) to 100 (highest image quality, lowest compression).

$$\mathbf{Q}_0 = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 84 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{4.4}$$

A quality-quantization evaluation was performed using the JPEG quantization factor, based on the matrix (4.4), using the same test used in CB-2017 (OLIVEIRA et al., 2017), but expanding to fifty test pictures of that same repository (USC-SIPI, 2017). The literature proposes only an isolated comparison methodology based on quality or entropy versus QP analysis (OLIVEIRA et al., 2015; OLIVEIRA et al., 2017; PAIM et al., 2017) that do not lead to an exact conclusion for comparison among the transforms. Therefore, this thesis proposes a straightforward quality versus spectral entropy to show a fair comparison of the quality losses for the same compression rate – represented by the entropy. All related works lack this analysis; hence a fair comparison cannot be made. Figure 4.2 presents three objective evaluation curves of compression efficiency with average quality versus average spectral entropy for (a) Peak signal-to-noise ratio (PSNR), (b) Structural Similarity Index (SSIM), and (c) Spectral residual-based similarity (SR-SIM). Therefore, the proposed analysis eliminates the QP dependency of the comparison.

Figure 4.2: Objective Compression Efficiency Comparison: (a) PSNR, (b)Y-SSIM, (c) SR-SIM, all vs. Average spectral entropy.



Source: The Author.

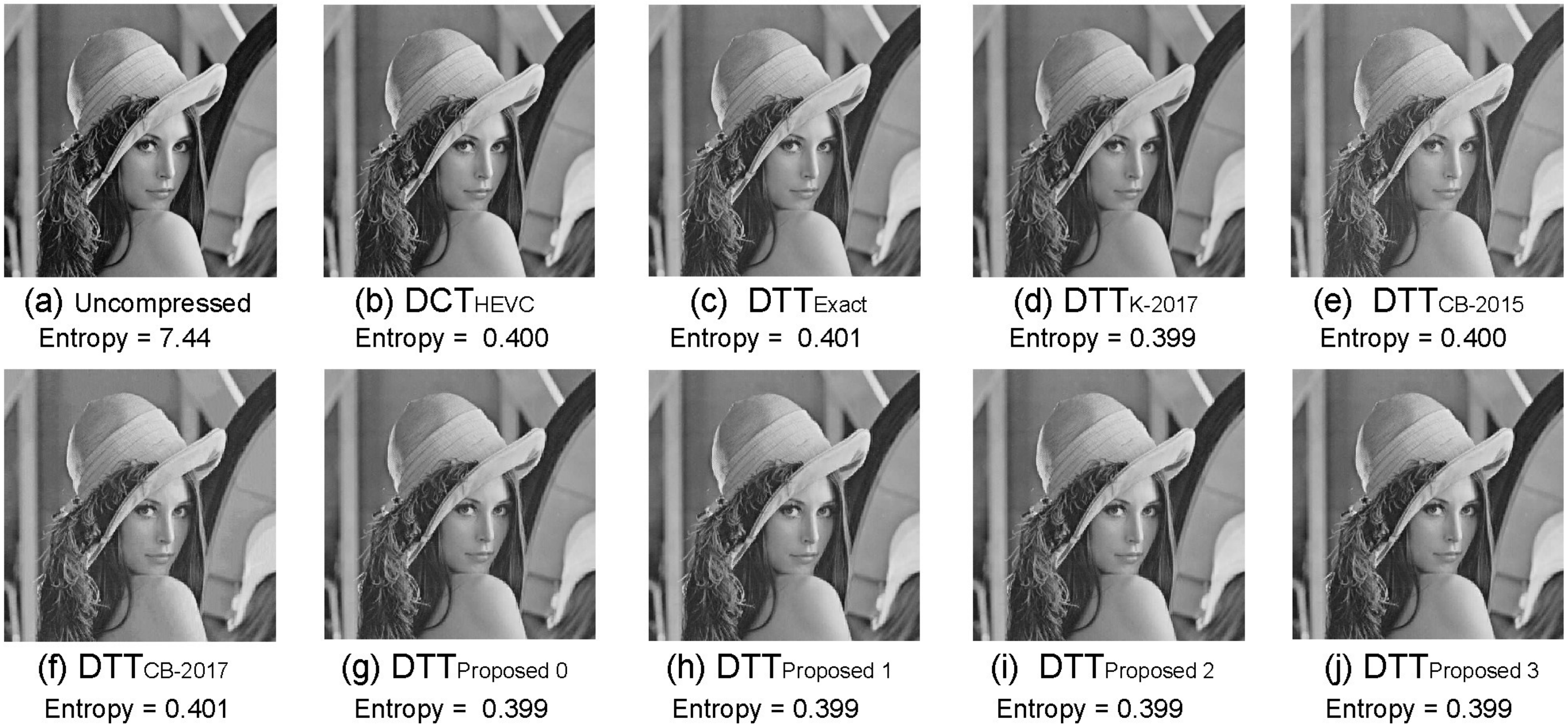
Regarding compression efficiency, Figure 4.2 shows that the DTT Proposal 0 approximation is better than the state-of-the-art, considering a compression point in about 0.4 of entropy region. Further, in Figure 4.2-b, it can be observed that in Proposal 1, the line removal improves compression efficiency compared with Proposal 0. This occurs because the compression efficiency reduces the entropy and the non-diagonal distortion, reducing the SSIM impact.

4.2 Subjective Image Quality Evaluation

Figure 4.3 presents a subjective quality analysis comparing Lena's uncompressed picture (Figure 4.3-a) with its compressed version, considering an entropy rate around 0.4 to maintain all transforms at approximately the same compression ratio. Thus, the resultant compressed image for each DCT and DTT evaluated is shown in: Figure 4.3-b) HEVC DCT (DO; TAN; YEO, 2014); c) Exact DTT (PRATTIPATI et al., 2013); d) K-2017 (KOUADRIA et al., 2017); e) CB-2015 DTT (OLIVEIRA et al., 2015); f) CB-2017 DTT (OLIVEIRA et al., 2017); g-j) DTT Proposal 0 to 3.

Aiming at subjective quality fair comparisons, our DTT Proposal architectures results, shown in Figure 4.3, were evaluated for spectral entropy operation points with almost, but less than 0.4. CB-2015 approximation (Figure 4.3-e) shows a severe loss of brightness and contrast. K-2017 (Figure 4.3-d) and CB-2017 (Figure 4.3-e) reduce the aforementioned losses. Our four proposals are shown in Figure 4.3-(g-j) add less noise than CB-2015, CB-2017, and K-2017 state-of-the-art 8-point DTT approximations with less than 0.4 spectral entropy rate in the subjective metric. Additionally, they also improve the brightness and contrast problem while keeping a better image quality than the approaches of CB-2015, especially in regions with lots of details such as the face, eyes, and Lena's hair.

Figure 4.3: Subjective quality comparison for almost 0.4 of entropy (a) Uncompressed, (b) 8-point HEVC DCT (DO; TAN; YEO, 2014), (c) Exact DTT (PRATTIPATI et al., 2013; PRATTIPATI; SWAMY; MEHER, 2013), (d) K-2017(KOUADRIA et al., 2017), (e) CB-2015, (f) CB-2017 (OLIVEIRA et al., 2017), (g) Proposal 0, (h) Proposal 1, (i) Proposed 2 and (j) Proposed 3.



(a) Uncompressed
Entropy = 7.44

(b) DCT$_{\text{HEVC}}$
Entropy = 0.400

(c) DTT$_{\text{Exact}}$
Entropy = 0.401

(d) DTT$_{\text{K-2017}}$
Entropy = 0.399

(e) DTT$_{\text{CB-2015}}$
Entropy = 0.400

(f) DTT$_{\text{CB-2017}}$
Entropy = 0.401

(g) DTT$_{\text{Proposed 0}}$
Entropy = 0.399

(h) DTT$_{\text{Proposed 1}}$
Entropy = 0.399

(i) DTT$_{\text{Proposed 2}}$
Entropy = 0.399

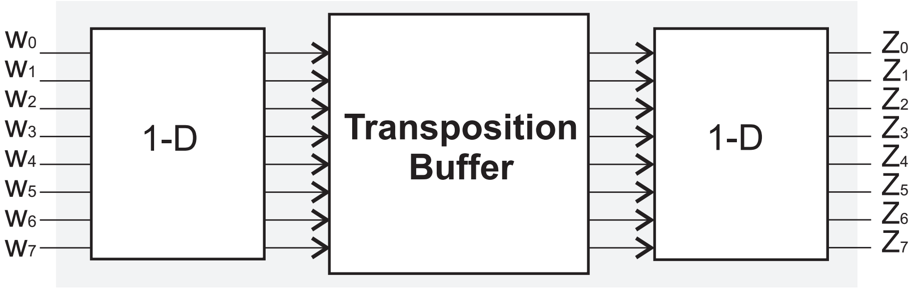
(j) DTT$_{\text{Proposed 3}}$
Entropy = 0.399

Source: The Author.

4.3 Truncated and Pruned Approximate DTT Hardware Design Proposals

The two-dimensional discrete transform can be realized following two different approaches: one needs transform blocks replication to calculate the entire matrix multiplication in a single cycle. Concurrently, the other uses the resource sharing principle to sequentially multiply matrices, reducing to only two transform blocks attached to a TB (Transposition Buffer). A high-level illustration of such hardware design is shown in Figure 4.4.

Figure 4.4: Transposition Buffer scheme.



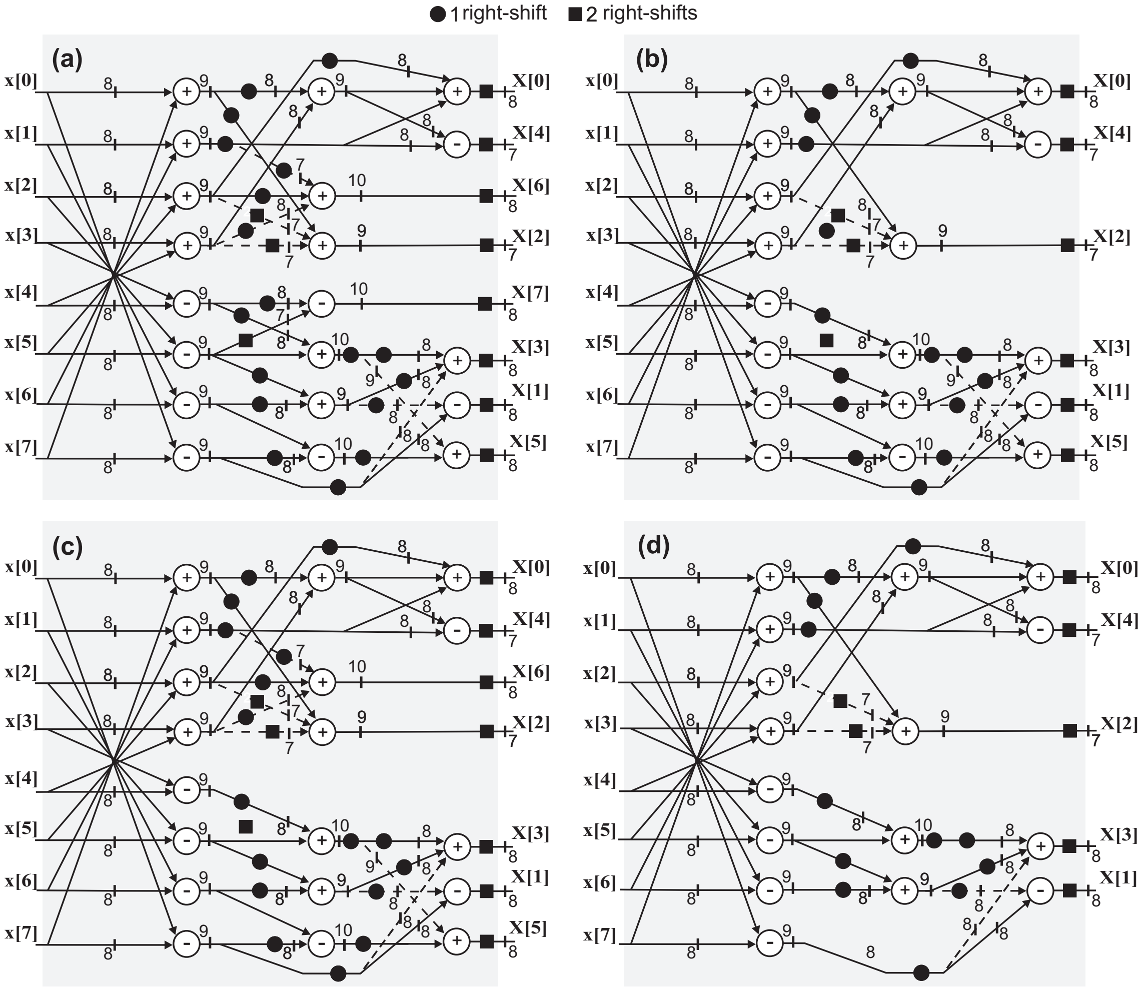
Source: The Author.

The 2-D DTT hardware design is based on the separability property of the DTT (NAKAGAKI; MUKUNDAN, 2007). The proposed approach uses two instances of 1-D 8-point DTT, which are separated by a transposition buffer. Figure 4.5 (a)-(d) presents the proposed 1-D 8-point approximate DTT hardware designs. The dashed arrows represent 2's complement representation. At each cycle, eight samples (one row) from an 8×8 block are read from the image (or residue) to feed the inputs of the 8-point DTT.

The truncation-only approach in Figure 4.5-a has twenty adders (four carry-save adders). This design follows the same approximation principle presented in the state-of-the-art 8-point DTT (OLIVEIRA et al., 2017). The main advantage of the proposed hardware design arises from the fractional coefficients of its base matrix (4.1), enabling the insertion of right-shifts. The right-shift-based truncation reduces both the non-orthogonality deviation error (demonstrated in Section III, Figure 1) and the bit-width in both the datapath and the transposition buffer. Further, this transform has 8-bit wide inputs, so the outputs have a maximum width of eight bits without increasing the bit-width of the second 1-D DTT stage. Since the output bit-width in (OLIVEIRA et al., 2017) is twelve bits, our proposed designs present transposition buffers with 33.3 % fewer bit-width when compared to the state-of-the-art designs.

Figure 4.5 (b), (c) and (d) show the proposed truncation with one-, two- and three-pruned rows, respectively. One can notice that as the pruning level increases, the number of outputs decreases accordingly. In the one-pruned row version in (b), the number of adders is nineteen. This number is further reduced to eighteen and sixteen in the two- and three-pruned rows proposed approximate 8-point 1-D DTT. Also, all three pruning-based approaches keep a maximum bit-width of eight bits.

Figure 4.5: 8-point 1-D proposed approximate DTT architectures.

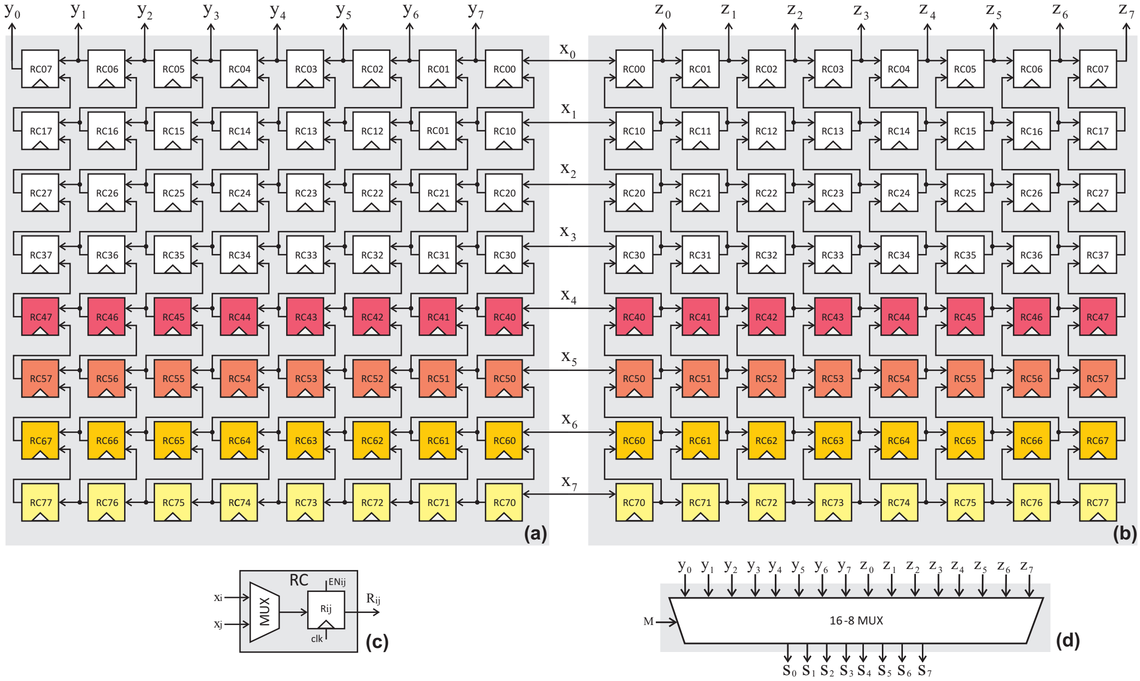


Source: The Author.

4.4 Efficient Pruned Transposition Buffer

A key factor in the hardware efficiency of the two-dimensional transform is how the 1-D transform approximations in the combinational logic impact the sequential elements in the transposition buffer used to connect both 1-D transform blocks. Thus, this section explores three different buffer architectures that allow simultaneous data reading and writing. An extensive analysis is herein performed to verify how the buffers can be

Figure 4.6: Transposition Buffer A - TB-A (CONCEIÇÃO et al., 2014).


y0 y1 y2 y3 y4 y5 y6 y7
z0 z1 z2 z3 z4 z5 z6 z7
x0 x1 x2 x3 x4 x5 x6 x7
RC07 RC06 RC05 RC04 RC03 RC02 RC01 RC00
RC17 RC16 RC15 RC14 RC13 RC12 RC01 RC10
RC27 RC26 RC25 RC24 RC23 RC22 RC21 RC20
RC37 RC36 RC35 RC34 RC33 RC32 RC31 RC30
RC47 RC46 RC45 RC44 RC43 RC42 RC41 RC40
RC57 RC56 RC55 RC54 RC53 RC52 RC51 RC50
RC67 RC66 RC65 RC64 RC63 RC62 RC61 RC60
RC77 RC76 RC75 RC74 RC73 RC72 RC71 RC70
RC00 RC01 RC02 RC03 RC04 RC05 RC06 RC07
RC10 RC11 RC12 RC13 RC14 RC15 RC16 RC17
RC20 RC21 RC22 RC23 RC24 RC25 RC26 RC27
RC30 RC31 RC32 RC33 RC34 RC35 RC36 RC37
RC40 RC41 RC42 RC43 RC44 RC45 RC46 RC47
RC50 RC51 RC52 RC53 RC54 RC55 RC56 RC57
RC60 RC61 RC62 RC63 RC64 RC65 RC66 RC67
RC70 RC71 RC72 RC73 RC74 RC75 RC76 RC77
(a)
(b)
RC
ENij
Xi
Xj
MUX
Rij
Rij
clk
(c)
y0 y1 y2 y3 y4 y5 y6 y7 z0 z1 z2 z3 z4 z5 z6 z7
M
16-8 MUX
S0 S1 S2 S3 S4 S5 S6 S7
(d)

Source: Adapted from Conceição et al. (2014).

tuned to optimize circuit power, area, and timing tradeoffs, when considering the simplifications, approximations, and coefficient pruning.

Figure 4.4 illustrates the buffer-based 2-D transform. As the first transform block calculates the multiplication, its results are stored in the TB. The buffer outputs the transposed version of the data initially stored according to the 2-D transform algorithm, fed to the final processing block.

The TB-A implements two-direction shift-registers. It aims to have a high throughput without the need for higher operating frequencies to compensate for the filling and emptying latencies of a traditional buffer. Such parallelism demands a two-stage buffer where when one stage is in write mode, receiving data from the first transform block, the second stage is in reading mode, sending data to the second transform block.

Figure 4.6 shows a straightforward implementation of such buffer (named TB-A) where each square box includes a mux and a register (see Fig. 4.6-c). This design may be seen as two 8×8 register arrays sharing the input interface, represented by the x_n wires. To keep both the inputs and outputs with the same bandwidth, the design features a multiplexer on its output to select the data source that corresponds to the array in the reading mode, shown in Figure 4.6-d.

To illustrate how TB-A works, let us assume that the array on the left (Fig. 4.6-a) is in write mode while the array on the right (Fig. 4.6-b) is in reading mode. The write enable signal is active for array 4.6-a and the mux control signal selects the outputs z$_n$ from the right bank. At each clock cycle, the left RC$_{i0}$ column registers the input data x$_n$ as the other columns update their values from the previous stages working as a first-in, first-out (FIFO) memory. Simultaneously, the bank in reading mode works in a row-wise fashion to transpose the data previously stored. Thus, when data is read from z$_n$, information from bottom lines is transmitted to top lines.

Pruning lines from the DTT matrix reduce the amount of data to be processed. Consequently, such pruning shall occur in the TB-A to eliminate unnecessary hardware. The colored boxes in Figure 4.6 show the storage elements that should be proportionally removed when the DTT matrix lines are eliminated. Therefore, if four lines are removed from the matrix, the buffer only needs half of its components.

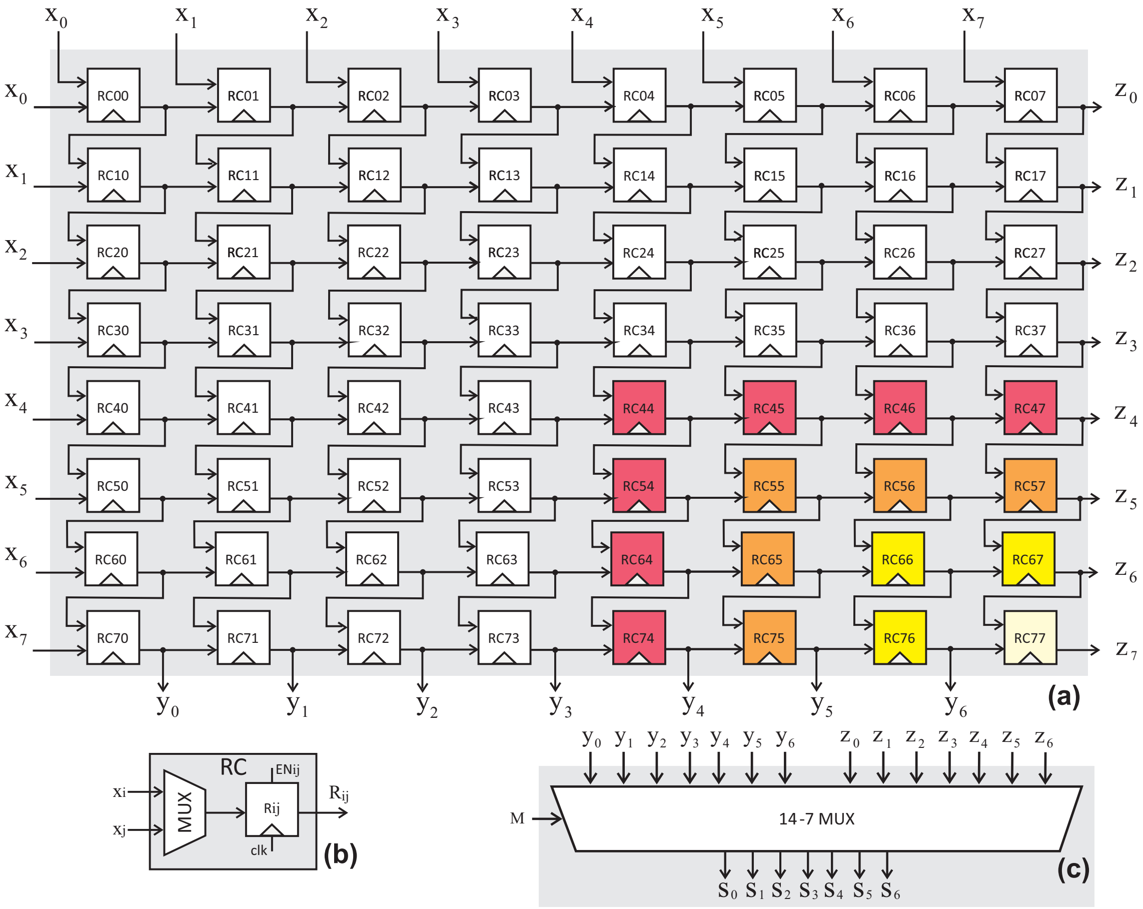
Carefully analyzing the array's timing in the reading mode in this architecture, it is clear that an entire line becomes unused after each cycle as the data is sent to the next stage. Hence, after eight clock cycles, the bank will be empty, leading to a low data occupation ratio considering the number of storage elements.

To reduce area and improve the occupation ratio, the architecture in Figure 4.7 proposes a single 8×8 register array, named TB-B, working with overlapped writing and reading states. In contrast to the previous design where the writing and reading mechanisms were out of phase, this TB operates in phase, i.e., both mechanisms access data either column- or row-wise. Similarly to TB-A, the input data enters TB-B either through the first column or row, updating the next storage elements following a wave pipeline approach. Thus, the outputs are both the last row and column, which are selected by the output mux (Fig. 4.7-c) upon the current data access mechanism.

This buffer works as follows: first, let us consider that TB-B has been filled row-wise, and, consequently, the reading has to be column-wise. As soon as the first data is read from y$_{7n}$, the entire buffer shifts its content one place to the right, freeing the leftmost column y$_{0n}$ that now may receive the output from the first transform. When the buffer is filled again, the data access mechanism is changed to row-wise, and the output mux selects the contents of the last row y$_{7n}$ and the information coming from x$_n$ is stored in the first line. This interchange occurs every eight cycles as long as the buffer is in use.

Due to the data access scheme's inherent interchange, the TB-B loses only one storage element per line following an L-pattern for each line pruned from the matrix. The colored boxes in Figure 4.7-(a) shows the pruning progression in this buffer from one matrix line (light yellow box) to four matrix lines (red boxes).

Figure 4.7: Transposition Buffer B - TB-B.

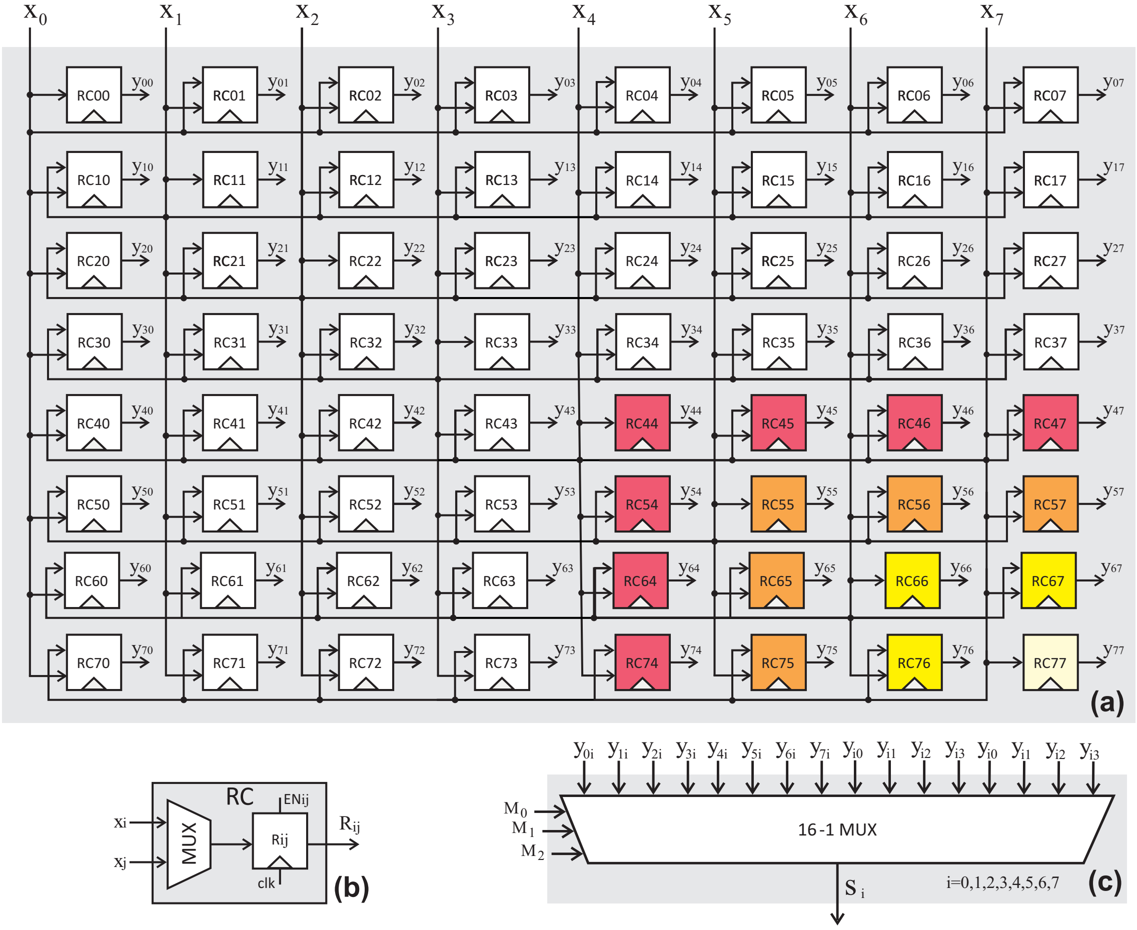

x0 x1 x2 x3 x4 x5 x6 x7
RC00 RC01 RC02 RC03 RC04 RC05 RC06 RC07
RC10 RC11 RC12 RC13 RC14 RC15 RC16 RC17
RC20 RC21 RC22 RC23 RC24 RC25 RC26 RC27
RC30 RC31 RC32 RC33 RC34 RC35 RC36 RC37
RC40 RC41 RC42 RC43 RC44 RC45 RC46 RC47
RC50 RC51 RC52 RC53 RC54 RC55 RC56 RC57
RC60 RC61 RC62 RC63 RC64 RC65 RC66 RC67
RC70 RC71 RC72 RC73 RC74 RC75 RC76 RC77
z0 z1 z2 z3 z4 z5 z6 z7
y0 y1 y2 y3 y4 y5 y6
(a)
RC
ENij
Xi
Xj
MUX
Rij
clk
Rij
(b)
y0 y1 y2 y3 y4 y5 y6 z0 z1 z2 z3 z4 z5 z6
M
14-7 MUX
S0 S1 S2 S3 S4 S5 S6
(c)

Source: The Author.

The FIFO-based implementation of buffer B has a high switching activity since all storage elements are updated to propagate data to the next stage for each read and write operation. To save dynamic power, Figure 4.8 proposes the TB-C, a slightly modified version of buffer B using element-wise addressing, which avoids such domino effect. Although it still switches between row- and column-wise data access, each line or column is directly accessed for the read/write operations. The design has eight 16-1 mux (Fig. 4.8-c) to select among the possible combinations of rows and columns to cope with this addressing scheme. Hence, this buffer has a minor area and routing overhead compared to the architecture described in Figure 4.7.

Since TB-C uses a different approach for data addressing, let us assume that the buffer has already been filled and that the current buffer access mechanism is row-wise. The output mux selects the contents of the last row y_{7n} and, subsequently, the output of the first transform x_n is stored in that same row. Next, the output mux selects the contents of row y_{6n}, and once again, the information coming from x_n is stored in that same spot. Once all rows have been processed, this procedure finishes, then the buffer switches to

Figure 4.8: Transposition Buffer C - TB-C (MEHER et al., 2014).


x0 x1 x2 x3 x4 x5 x6 x7
RC00 y00 RC01 y01 RC02 y02 RC03 y03 RC04 y04 RC05 y05 RC06 y06 RC07 y07
RC10 y10 RC11 y11 RC12 y12 RC13 y13 RC14 y14 RC15 y15 RC16 y16 RC17 y17
RC20 y20 RC21 y21 RC22 y22 RC23 y23 RC24 y24 RC25 y25 RC26 y26 RC27 y27
RC30 y30 RC31 y31 RC32 y32 RC33 y33 RC34 y34 RC35 y35 RC36 y36 RC37 y37
RC40 y40 RC41 y41 RC42 y42 RC43 y43 RC44 y44 RC45 y45 RC46 y46 RC47 y47
RC50 y50 RC51 y51 RC52 y52 RC53 y53 RC54 y54 RC55 y55 RC56 y56 RC57 y57
RC60 y60 RC61 y61 RC62 y62 RC63 y63 RC64 y64 RC65 y65 RC66 y66 RC67 y67
RC70 y70 RC71 y71 RC72 y72 RC73 y73 RC74 y74 RC75 y75 RC76 y76 RC77 y77
(a)
RC
Xi
Xj
MUX
ENij
Rij
clk
Rij
(b)
y0i y1i y2i y3i y4i y5i y6i y7i yi0 yi1 yi2 yi3 yi0 yi1 yi2 yi3
M0
M1
M2
16-1 MUX
S i
i=0,1,2,3,4,5,6,7
(c)

Source: The Author.

the column-wise operation mode. This interchange occurs every eight cycles as long as the buffer is in use. This buffer uses the same approach as buffer B to reduce the number of registers when pruning is used.

For each of the buffer architectures presented, a unique finite-state machine (FSM) was developed. Furthermore, a modified version of the unpruned buffer FSM was extended to support pruning for the various N-lines pruned architectures, resulting in fifteen different FSM implementations.

A shared characteristic of all implemented finite-state machines is that they perform the first complete operation cycle in sixteen clock cycles. This operation consists of transferring data from the first 1-D transform stage to the TB and feeding the second stage 1-D transform with the buffer's output data. The only difference among these FSM implementations resides in how and which registers are accessed according to which pruned architecture it is attached to.

Assuming the buffer is empty and used for the first time, the first eight clock cycles will fill the buffer with the first stage 1-D transform output data. The eight subsequent clock cycles will feed the second stage 1-D transform with the buffer's output data while also registering the input data according to each buffer architecture's inner-working mechanism. For the pruned designs, the first eight clock cycles will have the same behavior, recording the first transform's input data. However, the output data from the buffer to the second stage 1-D transform will be made available for the successive $8 - N$ clock cycles, where N is the number of pruned lines. For a two-lines pruned buffer, six clock cycles are necessary to feed the next stage transform. The remaining two clock cycles will have a high impedance output to reduce dynamic power. This asymmetry happens because, despite pruning the buffer, the input data will always need eight clock cycles to fill the buffer, leading to N unused clock cycles while outputting data.

4.5 Synthesis Results and Discussions

Our proposal approximate DTT, the related work approximate DTT, and the HEVC DCT architectures were described in Verilog HDL with the same parallelism level, processing eight coefficients per cycle. The circuits were synthesized for both ASIC and FPGA hardware technologies. The RTL description for the related work architectures was implemented considering the same factorization and electrical schemes, according to each reference. For a fair comparison, every architecture was synthesized under the same conditions and using the same synthesis methodology for ASIC and FPGA digital flows. The target frequency was calculated in order to sustain the minimum required throughput to process 4k (3840×2160p) videos at 30 frames per second, considering a 4:2:0 sub-sampling (sub=1.5). Equation 4.5 shows the minimum frequency target calculation. The numerator represents the total amount of pixels in one second, and the denominator the number of pixels processed in one cycle.

$$F_{Target} = \frac{Resolution * fps * sub}{parallelism} \tag{4.5}$$

4.5.1 ASIC Synthesis Results

The synthesis for ASIC was performed in Cadence RTL Compiler tool (CADENCE, 2020a) using the 65nm ST standard cell library (ST, 2013) at 1.0V voltage supply. According to Equation 4.5, for 4k@30fps videos with a 4:2:0 sub-sampling, our architecture needs to be able to process at least 373 Mpixels per second. An operating frequency of 46.7MHz is the minimum necessary, and we synthesize with a target for 50MHz to include a timing guardband to guarantee the performance after the physical synthesis.

Firstly, a synthesis in RTL Compiler™ is performed to generate the Verilog netlist and the SDF delay file. After that, the simulation generates the VCD stimuli file and presents a testbench composed of MATLAB™ and Cadence Incisive™, considering real input vectors of four test images: lena, boat, airplane, and baboon. Each image contains 512×512 luminance pixels, and therefore, the test includes 32768 lines of eight pixels samples.

Table 4.1: Related Work Power Extraction Flow.

	Stages of Accurately Power Extraction Flow						
Related Work	Real Vectors	Netlist Simul.	.SDF file	.VCD or .SAIF file	Cap. Table	PLE	Ind. Lib
(OLIVEIRA et al., 2015)	×	×	×	×	×	×	×
(OLIVEIRA et al., 2017)	×	✓	×	×	×	✓	✓
(PAIM et al., 2017)	✓	✓	✓	✓	×	×	×
(KOUADRIA et al., 2017)	×	×	×	×	×	×	×
Ours	✓	✓	✓	✓	✓	✓	✓

Cadence logic synthesis tool (CADENCE, 2020a) also allows an option for the interconnection estimation mode called physically-aware layout estimation (PLE) mode. This method estimates the length of the nets and takes into account the load capacitance effects in the power dissipation, considering a relatively pessimistic layout estimation. The PLE mode also requires the inclusion of the library exchange format (LEF) files, mainly containing the library's physical layout information. A realistic power extraction is done using an industrial standard cells library (Ind. Lib), such as the ST 65nm (ST, 2013) used in this work. The LEF macro includes the inner library cell's capacitance, and the tech LEF comprises the process metal capacitance for the interconnection capacitance estimation (CADENCE, 2020b). The capacitance table file (CapTable) contains the same information as the LEF but in a more precise and fine-grained way, as it considers the process variations (CADENCE, 2020b).

Although many works in the literature have proposed DTT architectures, most of them do not report power dissipation results, as shown in Table 3.1. Some works exploit the use of approximations in DTT, but not within a realistic test environment. Power dissipation is estimated for all the DTT architectures versions with realistic input data from four different test images. Table 4.1 summarizes the various inputs needed for a realistic power dissipation extraction methodology and shows how the related works use them.

Tables 4.2 and 4.3 present the ASIC synthesis results for timing, circuit area, and power dissipation, respectively. The results include all evaluated transforms and the three different TB architectures TB-A, TB-B, and TB-C. The pruning of the TBs follows the transform prune.

Table 4.2 shows the CPD and maximum frequency (Max. Freq.) results. Proposal 0 approximation is 9.5% (TB-A), 18.1% (TB-B) and 11.6% (TB-C) faster than the state-of-the-art 8-point DTT CB-2017 (OLIVEIRA et al., 2017), and 51.8% (TB-A), 85.1% (TB-B) and 40.2% (TB-C) when compared with 8-point DCT of the HEVC standard (MCCANN et al., 2013). The gains demonstrated by our solutions are due to the reduction of the bit-width caused by the proposed truncation technique implemented with internal right-shifts. The variation on the maximum frequency on the proposed DTTs is an effect of the logic synthesis tool. It was set to prioritize area and power optimization, which may lead to a longer CPD due to the logical topology's choice to compute the logic function. Despite this factor, almost all proposed architectures presented significant improvement on the maximum frequency.

DTTs, when using the TB-B, have proven to be the ones with the highest maximum frequency among all implemented architectures. The longer CPD on TB-A is due to the increased fan-out on the 1-D transform necessary to load the dual register bank. An advantage of TB-B over TB-C is the reduced mux usage as it uses seven 2-1 mux in parallel – only one mux in the CPD – instead of the eight 16-1 mux required by the latter – four 2-1 mux in the CPD. While the CPD for the Proposal 3 architecture is $10576ps$ and $8939ps$ for TB-B and TB-C, respectively, the CPD for TB-B is only $7689ps$. When using TB-B, the Proposal 3 architecture achieves a maximum frequency of 53.2% higher than that of K-2017 DTT.

Table 4.2: ASIC circuit area, critical path delay, and maximum frequency results.

Circuit		Gate Count	Cell Area	Area Variation (%)		CPD	Max. Freq.	Max. Freq. Variation (%)		Max. Throughput
Version	TB	(kgates)	(μm^2)	**CB-2017**	**K-2017**	(ps)	(MHz)	**CB-2017**	**K-2017**	($Mpixels/s$)
$\textbf{DCT}_{\textbf{HEVC}}$	A	14196	29527	58.5	90.8	14830	67.4	-27.8	-9.5	539.2
$\textbf{DTT}_{\textbf{Exact}}$	A	12797	26617	42.9	72.0	13810	72.4	-22.5	-2.8	579.2
$\textbf{DTT}_{\textbf{CB-2015}}$	A	11076	23039	23.7	48.9	12799	78.1	-16.4	4.8	624.8
$\textbf{DTT}_{\textbf{CB-2017}}$	A	8957	18630	-	20.4	10701	93.4	-	25.4	747.2
$\textbf{DTT}_{\textbf{K-2017}}$	A	7440	15475	-16.9	-	13418	74.5	-20.2	-	596.0
$\textbf{DTT}_{\textbf{Proposal 0}}$	A	6736	14011	-24.8	-9.5	9774	102.3	9.5	37.3	818.4
$\textbf{DTT}_{\textbf{Proposal 1}}$	A	6931	14417	-22.6	-6.8	9636	103.8	11.1	39.3	110.4
$\textbf{DTT}_{\textbf{Proposal 2}}$	A	6015	12512	-32.8	-19.2	8659	115.5	23.7	55.0	924.0
$\textbf{DTT}_{\textbf{Proposal 3}}$	A	5176	10767	-42.2	-30.4	10576	94.6	1.3	26.9	756.8
$\textbf{DCT}_{\textbf{HEVC}}$	B	11011	22902	82.3	74.3	13132	76.1	-36.2	-10.4	608.8
$\textbf{DTT}_{\textbf{Exact}}$	B	9564	19894	58.4	51.4	11515	86.8	-27.2	2.2	694.4
$\textbf{DTT}_{\textbf{CB-2015}}$	B	7829	16285	29.7	23.9	10044	99.6	-16.5	17.3	796.8
$\textbf{DTT}_{\textbf{CB-2017}}$	B	6038	12560	-	-4.4	8384	119.3	-	40.5	954.4
$\textbf{DTT}_{\textbf{K-2017}}$	B	6318	13141	4.6	-	11784	84.9	-28.8	-	679.2
$\textbf{DTT}_{\textbf{Proposal 0}}$	B	4394	9139	-27.2	-30.5	7097	140.9	18.1	66.0	1127.2
$\textbf{DTT}_{\textbf{Proposal 1}}$	B	4433	9221	-26.6	-29.8	7474	133.8	12.2	57.6	1070.4
$\textbf{DTT}_{\textbf{Proposal 2}}$	B	4217	8771	-30.2	-33.3	7475	133.8	12.2	57.6	1070.4
$\textbf{DTT}_{\textbf{Proposal 3}}$	B	3837	7980	-36.5	-39.3	7689	130.1	9.0	53.2	1040.8
$\textbf{DCT}_{\textbf{HEVC}}$	C	13510	28100	58.5	91.7	17965	55.7	-20.5	-32.5	445.6
$\textbf{DTT}_{\textbf{Exact}}$	C	10577	22000	24.1	50.1	15673	63.8	-8.9	-22.7	510.4
$\textbf{DTT}_{\textbf{CB-2015}}$	C	12234	25446	43.5	73.6	17405	57.5	-17.9	-30.3	460.0
$\textbf{DTT}_{\textbf{CB-2017}}$	C	8524	17730	-	21.0	14271	70.1	-	-15.0	560.8
$\textbf{DTT}_{\textbf{K-2017}}$	C	7046	14655	-17.3	-	12120	82.5	17.7	-	660.0
$\textbf{DTT}_{\textbf{Proposal 0}}$	C	6476	13470	-24.0	-8.1	12810	78.1	11.6	-5.3	624.8
$\textbf{DTT}_{\textbf{Proposal 1}}$	C	6001	12482	-29.6	-14.8	8719	114.7	63.6	39.0	917.6
$\textbf{DTT}_{\textbf{Proposal 2}}$	C	5465	11367	-35.9	-22.4	8696	115.0	64.0	39.4	920.0
$\textbf{DTT}_{\textbf{Proposal 3}}$	C	4809	10003	-43.6	-31.7	8939	111.9	59.6	35.6	895.2

Table 4.3: Power dissipation results for ASIC @ 50 MHz operating frequency target.

Circuit		Static	Dynamic	Total Power Dissipation			Total Power Variation (%)					
Version	TB	$\overline{x}$ (μW)	$\overline{x}$ (μW)	$\overline{x}$ (μW)	σ	C_V (%)	**DCT$_{\textbf{HEVC}}$**	**DTT$_{\textbf{Exact}}$**	**DTT$_{\textbf{CB-2015}}$**	**DTT$_{\textbf{CB-2017}}$**	**DTT$_{\textbf{K-2017}}$**	**DTT$_{\textbf{Proposal 0}}$**
DCT$_{\textbf{HEVC}}$	A	14.2	736.7	750.9	9.5	1.3	-	8.3	32.6	42.7	119.5	79.2
DTT$_{\textbf{Exact}}$	A	13.1	680.0	693.1	10.8	1.6	-7.7	-	22.4	29.3	102.6	65.4
DTT$_{\textbf{CB-2015}}$	A	11.7	554.6	566.4	3.9	0.7	-24.6	-18.3	-	3.5	65.6	35.2
DTT$_{\textbf{CB-2017}}$	A	9.6	526.3	535.9	4.6	0.9	-28.6	-22.7	-5.4	-	56.7	27.9
DTT$_{\textbf{K-2017}}$	A	7.1	335.0	342.1	5.5	1.6	-54.4	-50.6	-39.6	-36.1	-	-18.3
DTT$_{\textbf{Proposal 0}}$	A	7.3	411.7	419.0	5.6	1.3	-44.2	-39.5	-26.0	-21.8	22.5	-
DTT$_{\textbf{Proposal 1}}$	A	7.1	355.5	362.6	5.5	1.5	-51.7	-47.6	-35.9	-32.3	6.0	-13.5
DTT$_{\textbf{Proposal 2}}$	A	6.1	305.8	312.0	4.6	1.5	-58.4	-54.9	-44.9	-41.7	-8.8	-25.6
DTT$_{\textbf{Proposal 3}}$	A	5.2	254.6	259.8	4.2	1.6	-65.4	-62.5	-54.1	-51.5	-24.0	-37.9
DCT$_{\textbf{HEVC}}$	B	10.6	518.9	529.5	9.6	1.8	-	12.0	53.0	57.6	68.9	104.2
DTT$_{\textbf{Exact}}$	B	9.4	463.5	472.9	11.0	2.3	-10.6	-	34.0	40.8	47.8	78.7
DTT$_{\textbf{CB-2015}}$	B	7.8	338.2	346.0	4.1	1.2	-34.6	-26.8	-	3.0	10.4	33.4
DTT$_{\textbf{CB-2017}}$	B	6.3	329.6	335.9	4.7	1.4	-36.5	-28.9	-2.9	-	7.1	29.5
DTT$_{\textbf{K-2017}}$	B	6.2	307.2	313.5	5.3	1.7	-40.8	-33.7	-9.40	-6.6	-	20.9
DTT$_{\textbf{Proposal 0}}$	B	4.6	254.7	259.3	5.3	2.1	-51.0	-45.1	-25.0	-22.7	-17.3	-
DTT$_{\textbf{Proposal 1}}$	B	4.7	245.1	249.8	5.4	2.2	-52.8	-47.1	-27.8	-25.6	-20.3	-3.7
DTT$_{\textbf{Proposal 2}}$	B	4.4	232.4	236.8	4.7	2.0	-55.2	-49.9	-31.5	-29.4	-24.4	-8.7
DTT$_{\textbf{Proposal 3}}$	B	4.0	211.3	215.3	4.3	2.0	-59.3	-54.4	-37.7	-35.9	-31.3	-16.9
DCT$_{\textbf{HEVC}}$	C	11.8	498.7	510.5	9.9	1.9	-	57.6	13.9	61.0	75.7	109.5
DTT$_{\textbf{Exact}}$	C	9.3	314.6	324.0	5.2	1.6	-36.5	-	-27.7	2.2	11.5	32.9
DTT$_{\textbf{CB-2015}}$	C	10.8	437.3	448.1	11.7	2.6	-12.2	38.3	-	41.4	54.2	83.9
DTT$_{\textbf{CB-2017}}$	C	7.6	309.3	317.0	5.2	1.6	-37.9	-2.2	-29.3	-	9.1	30.1
DTT$_{\textbf{K-2017}}$	C	6.3	284.3	290.6	6.1	2.1	-43.1	-10.3	-35.1	-8.3	-	19.2
DTT$_{\textbf{Proposal 0}}$	C	5.7	238.0	243.7	8.0	3.3	-52.3	-24.8	-45.6	-23.1	-16.1	-
DTT$_{\textbf{Proposal 1}}$	C	5.4	231.0	236.4	7.4	3.1	-53.7	-27.0	-47.2	-25.4	-18.6	-3.0
DTT$_{\textbf{Proposal 2}}$	C	5.0	220.7	225.7	6.6	2.9	-55.8	-30.3	-49.6	-28.8	-22.3	-7.4
DTT$_{\textbf{Proposal 3}}$	C	4.4	198.0	202.3	6.2	3.1	-60.4	-37.6	-54.8	-36.2	-30.4	-17.0

Comparative circuit area results are shown in Table 4.2 using two metrics: gate count and the physical circuit cell area. The gate count is given in the area-equivalent number of 2-input NANDs. The proposed approximations offer massive area reduction ranging from 22.6% (Proposal 1, TB-A) up to 43.6% (Proposal 3, TB-C) compared to the CB-2017 2-D transform. Similar gains are observed compared to K-2017 transforms, from 6.8% (Proposal 1, TB-A) up to 39.3% (Proposal 3, TB-B). In all cases, TB-B is the most area-efficient structure, reaching savings of 27.9% and 22.8% when compared to TB-A and TB-C designs, respectively.

Table 4.3 shows the average ($\overline{x}$), standard deviation (σ), and the coefficient of variation (C_V) for the total power for the HEVC DCT (BUDAGAVI et al., 2013) and all DTT versions. C_V between 0.7 % and 3.3 % means that the total power dissipation values for the images were very close to the average. The exact baseline DTT presents gains over the HEVC DCT of 18%, on average, considering all buffer topologies. Therefore, any improvements made over the exact DTT would surpass the HEVC DCT when aiming for low-power operation. Using our pruning and truncation approach, our proposed hardware designs led to significant power dissipation savings compared to state-of-the-art DTT hardware designs.

Table 4.3 also presents a comparison of total power dissipation between each literature version. This comparison demonstrates that our proposal techniques provide in the 8-point DTT Proposal 3 version 65.4% less power than the HEVC standard DCT, and it outperforms both CB-2017 and K-2017 state-of-the-art 8-point DTTs with 51.5% and 24.0% less power dissipation – considering the TB-A topology.

Another essential aspect that impacts the results is the different buffer mechanisms. Compared to TB-A, TB-B and TB-C present average power savings of 28.5% and 32.0%, respectively. This fact arises from the smaller number of registers and, consequently, better utilization of memory cells and the reduced glitching activity due to the data access strategies.

Our analysis also shows that TB-C is the most power-efficient buffer. Compared with TB-B – the most area-efficient buffer – TB-C shows 4.5% savings in total power dissipation, even with more circuit area. Despite having a higher input capacitance, the TB-C hardware design compensates this with power dissipation savings due to the data addressing scheme that allows writing each row/column independently, causing switching activity in only eight registers per cycle. This TB-C approach avoids the switching caused by the domino effect of TB-B and TB-A.

4.5.2 FPGA Synthesis Results

Each hardware architecture was implemented and synthesized for FPGA using the Xilinx Vivado™ Design Suite (XILINX, 2019) targeting the Xilinx Virtex-7 XC7VX1140TF FPGA device, which is fabricated in the 28 nm CMOS process. For a precise power estimation, all simulations were performed on the netlist generated at the end of the FPGA device implementation flow, similar to the ASIC flow. It comprehends logic synthesis, technology mapping, placement, and routing. This post-implementation netlist, alongside a testbench that considers real input vectors of the same four test images (Lena, boat, airplane, and baboon), were used to generate the SAIF file, which is the only circuit activity file dump accepted for power estimation in Xilinx Power Analyser™. The synthesis results include all evaluated transforms for the three different TB architectures (TB-A, TB-B, and TB-C). The pruning of the TBs follows the pruning used in each transform.

Table 4.4 shows the circuit area (measured in the number of logic/register slices), CPD, and maximum frequency (Max. Freq.) FPGA results. Architectures using transposition buffer B have the smallest area and, on average, are 22.62% smaller than those using TB-A and 7.77% smaller when compared to TB-C architectures. Further, all of our proposed architectures are smaller than CB-2017 (OLIVEIRA et al., 2017) with savings of 22.7% for TB-A, 21.9% for TB-B, and 16.6% for TB-C. When pruning is considered, the Proposal 3 version outperforms all other architectures with area reduction up to 46.3% when compared to CB-2017 and up to 28.7% when compared to K-2017 (KOUADRIA et al., 2017).

Similarly, transposition buffer B-based transforms offer the highest maximum frequencies among the implemented architectures. Due to both the different multiplexers used in each circuit and the routing complexity of each architecture, TB-B architectures are 21.7% faster than TB-A circuits and 33.0% faster than TB-C architectures on average.

Table 4.4: FPGA circuit area, critical path delay, and maximum frequency results.

Circuit		Slices			Slice Variation (%)		CPD	Max. Freq.	Max. Freq. Variation (%)		Max. Throughput
Version	TB	LUTs	Registers	Total	**CB-2017**	**K-2017**	(ns)	(MHz)	**CB-2017**	**K-2017**	($Mpixels/s$)
$\mathbf{DCT_{HEVC}}$	A	1264	1504	2768	12.7	61.0	12.949	77.22	-31.8	-23.2	617.8
$\mathbf{DTT_{Exact}}$	A	1509	1576	3085	25.6	79.5	9.650	103.63	-8.4	3.1	829.0
$\mathbf{DTT_{CB\text{-}2015}}$	A	1694	1584	3278	33.5	90.7	10.203	98.01	-13.4	-2.5	784.1
$\mathbf{DTT_{CB\text{-}2017}}$	A	1025	1431	2456	-	42.9	8.838	113.15	-	12.6	905.2
$\mathbf{DTT_{K\text{-}2017}}$	A	891	828	1719	-30.0	-	9.949	100.51	-11.2	-	804.1
$\mathbf{DTT_{Proposal\ 0}}$	A	762	1136	1898	-22.7	10.4	7.168	139.51	23.3	38.8	1116.1
$\mathbf{DTT_{Proposal\ 1}}$	A	883	1016	1899	-22.7	10.5	7.951	125.78	11.2	25.1	1006.2
$\mathbf{DTT_{Proposal\ 2}}$	A	674	896	1570	-36.1	-8.7	7.271	137.53	21.5	36.8	1100.2
$\mathbf{DTT_{Proposal\ 3}}$	A	576	744	1320	-46.3	-23.2	6.599	151.54	33.9	50.8	1212.3
$\mathbf{DCT_{HEVC}}$	B	1434	1248	2682	57.3	72.4	10.191	98.13	-13.9	-19.7	785.0
$\mathbf{DTT_{Exact}}$	B	1297	960	2257	32.4	45.1	8.184	122.19	7.3	0.0	977.5
$\mathbf{DTT_{CB\text{-}2015}}$	B	1322	1000	2322	36.2	49.2	6.783	147.43	29.4	20.7	1179.4
$\mathbf{DTT_{CB\text{-}2017}}$	B	834	871	1705	-	9.6	8.778	113.92	-	-6.8	911.4
$\mathbf{DTT_{K\text{-}2017}}$	B	816	740	1556	-8.7	-	8.185	122.17	7.2	-	977.4
$\mathbf{DTT_{Proposal\ 0}}$	B	611	720	1331	-21.9	-14.5	6.122	163.35	43.4	33.7	1306.8
$\mathbf{DTT_{Proposal\ 1}}$	B	587	702	1289	-24.4	-17.2	6.455	154.92	36.0	26.8	1239.4
$\mathbf{DTT_{Proposal\ 2}}$	B	547	672	1219	-28.5	-21.7	5.852	170.88	50.0	39.9	1367.0
$\mathbf{DTT_{Proposal\ 3}}$	B	490	620	1110	-34.9	-28.7	5.525	181.00	58.9	48.2	1448.0
$\mathbf{DCT_{HEVC}}$	C	1806	1088	2894	56.3	84.8	14.582	68.58	-18.8	-33.4	548.6
$\mathbf{DTT_{Exact}}$	C	1532	864	2396	29.4	53.0	11.372	87.94	4.1	-14.5	703.5
$\mathbf{DTT_{CB\text{-}2015}}$	C	1635	879	2514	35.7	60.5	11.256	88.84	5.1	-13.7	710.7
$\mathbf{DTT_{CB\text{-}2017}}$	C	1070	782	1852	-	18.3	11.833	84.51	-	-17.9	676.1
$\mathbf{DTT_{K\text{-}2017}}$	C	838	728	1566	-15.4	-	9.718	102.90	21.8	-	823.2
$\mathbf{DTT_{Proposal\ 0}}$	C	913	632	1545	-16.6	-1.3	10.726	93.23	10.3	-9.4	745.8
$\mathbf{DTT_{Proposal\ 1}}$	C	811	677	1488	-19.7	-5.0	7.903	126.53	49.7	23.0	1012.2
$\mathbf{DTT_{Proposal\ 2}}$	C	734	648	1382	-25.4	-11.7	6.660	150.15	77.7	45.9	1201.2
$\mathbf{DTT_{Proposal\ 3}}$	C	541	596	1137	-38.6	-27.4	6.448	155.09	83.5	50.7	1240.7

All proposed architectures offer impressive maximum frequency improvements over the state-of-the-art DTTs. Except for the Proposal 0 version using TB-C, which has a slightly lower frequency than the one achieved by K-2017 using the same buffer, these frequency gains range from 10.3% (Proposal 0 vs. CB-2017, TB-A) to 83.5% (Proposal 3 vs. CB-2017, TB-C). As with the ASIC results, FPGA results have demonstrated that our proposed solutions surpass the ones present in the literature, both concerning quality and circuit operation – area, operating frequency, and power.

Table 4.5 shows the average ($\overline{x}$), standard deviation (σ), and the coefficient of variation (C_V) for the dynamic power for each HEVC DCT (BUDAGAVI et al., 2013) and all DTT versions. C_V between 1.5% and 6.9% means that the different images' dynamic power dissipation values were very close to the average. When mapped to FPGA, our proposed DTT hardware designs, with the pruning and truncation approach, also led to significant power dissipation savings compared to the state-of-the-art DTT hardware design.

Proposal 0 approximation dissipates 35.6% (TB-A), 41.2% (TB-B) and 37.3% (TB-C) less power than the state-of-the-art 8-point DTT CB-2017 (OLIVEIRA et al., 2017); and 56.1% (TB-A), 68.6% (TB-B) and 66.5% (TB-C) when compared with 8-point DCT of the HEVC standard (MCCANN et al., 2013). As with the ASIC results, FPGA results have demonstrated that our proposed solutions are better than the ones present in the literature. In contrast to the ASIC power dissipation results, which are in $micro$watt, the FPGAs results are in the order of $mili$watts. This is due to the reconfigurable characteristic of SRAM-based FPGAs, which offers excellent flexibility but sacrifices power. Static power is directly proportional to the number of transistors in the circuit. Since FPGAs have a fixed – and high – a number of the transistor, static power stays relatively the same for all architectures, while the ASIC results show that static power lowers as the circuit area is reduced.

Table 4.5: Power synthesis results for FPGA @ 50 MHz operating frequency target.

Circuit		Static	Dynamic Power Dissipation			Dynamic Power Variation (%)					
Version	TB	$\overline{x}$ (mW)	$\overline{x}$ (mW)	σ	C_V (%)	**DCT$_{\text{HEVC}}$**	**DTT$_{\text{Exact}}$**	**DTT$_{\text{CB-2015}}$**	**DTT$_{\text{CB-2017}}$**	**DTT$_{\text{K-2017}}$**	**DTT$_{\text{Proposal 0}}$**
DCT$_{\text{HEVC}}$	A	590.0	131.0	2.4	1.9	-	19.6	6.1	46.8	223.5	127.8
DTT$_{\text{Exact}}$	A	590.0	109.5	2.5	2.3	-16.4	-	-11.3	22.7	170.4	90.4
DTT$_{\text{CB-2015}}$	A	590.0	123.5	1.9	1.6	-5.7	12.8	-	38.4	204.9	114.8
DTT$_{\text{CB-2017}}$	A	590.0	89.3	2.1	2.3	-31.9	-18.5	-27.7	-	120.4	55.2
DTT$_{\text{K-2017}}$	A	589.0	40.5	1.7	4.3	-69.1	-63.0	-67.2	-54.6	-	-29.6
DTT$_{\text{Proposal 0}}$	A	589.0	57.5	2.6	4.6	-56.1	-47.5	-53.4	-35.6	42.0	-
DTT$_{\text{Proposal 1}}$	A	589.0	47.0	2.9	6.3	-64.1	-57.1	-61.9	-47.3	16.0	-18.3
DTT$_{\text{Proposal 2}}$	A	589.0	35.5	1.7	4.9	-72.9	-67.6	-71.3	-60.2	-12.3	-38.3
DTT$_{\text{Proposal 3}}$	A	589.0	26.3	1.3	4.8	-80.0	-76.0	-78.7	-70.6	-35.2	-54.3
DCT$_{\text{HEVC}}$	B	590.3	160.0	4.1	2.6	-	53.1	48.1	87.1	329.5	218.4
DTT$_{\text{Exact}}$	B	590.0	104.5	2.5	2.4	-34.7	-	-3.2	22.2	180.5	108.0
DTT$_{\text{CB-2015}}$	B	590.0	108.0	1.6	1.5	-32.5	3.3	-	26.3	189.9	114.9
DTT$_{\text{CB-2017}}$	B	590.0	85.5	1.7	2.0	-46.6	-18.2	-20.8	-	129.5	70.1
DTT$_{\text{K-2017}}$	B	589.0	37.3	1.3	3.4	-76.7	-64.4	-65.5	-56.4	-	-25.9
DTT$_{\text{Proposal 0}}$	B	589.0	50.3	2.6	5.2	-68.6	-51.9	-53.5	-41.2	34.9	-
DTT$_{\text{Proposal 1}}$	B	589.0	44.3	2.6	5.9	-72.3	-57.7	-59.0	-48.2	18.8	-11.9
DTT$_{\text{Proposal 2}}$	B	589.0	34.5	1.7	5.0	-78.4	-67.0	-68.1	-59.6	-7.4	-31.3
DTT$_{\text{Proposal 3}}$	B	589.0	25.3	1.3	5.0	-84.2	-75.8	-76.6	-70.5	-32.2	-49.8
DCT$_{\text{HEVC}}$	C	591.0	218.5	6.6	3.0	-	42.3	41.2	87.2	288.4	198.3
DTT$_{\text{Exact}}$	C	590.0	153.5	4.2	2.7	-29.7	-	-0.8	31.5	172.9	109.6
DTT$_{\text{CB-2015}}$	C	590.0	154.8	3.3	2.1	-29.2	0.8	-	32.5	175.1	111.3
DTT$_{\text{CB-2017}}$	C	590.0	116.8	4.6	3.9	-46.6	-23.9	-24.6	-	107.6	59.4
DTT$_{\text{K-2017}}$	C	589.0	56.3	1.3	2.2	-74.3	-63.4	-63.7	-51.8	-	-23.2
DTT$_{\text{Proposal 0}}$	C	589.3	73.3	5.0	6.8	-66.5	-52.3	-52.7	-37.3	30.2	-
DTT$_{\text{Proposal 1}}$	C	589.0	71.3	4.2	5.9	-67.4	-53.6	-54.0	-39.0	26.7	-2.7
DTT$_{\text{Proposal 2}}$	C	589.0	56.0	2.9	5.3	-74.4	-63.5	-63.8	-52.0	-0.4	-23.5
DTT$_{\text{Proposal 3}}$	C	589.0	36.5	2.5	6.9	-83.3	-76.2	-76.4	-68.7	-35.1	-50.2

4.6 Conclusion

This section presented an efficient approximate 8-point DTT hardware design combining pruning and approximation of coefficients. The key results showed that our proposal of suitable approximations in the coefficients and of appropriate pruning in the lines of the matrix, enabled by our 8-point DTT designed solutions, resulted in a maximum clock frequency increase of about 64%, up to 43.6% savings in standard cell logic area, and up to 65.4% power dissipation savings for ASIC implementation. Considering the FPGA mapping, the results show an increase of 83.4% in maximum clock frequency, 46.3% area reduction, and up to 84.2% power dissipation savings. Furthermore, our newer approximation in the hardware design also improves the DTT compression ratio with less quality degradation in the compressed image when compared with state-of-the-art approximate 8-point DTT hardware designs. The design exploration of transposition buffers indicated that the TB-B is the best-suited topology for area-constrained designs, while TB-C represents the most power- efficient design, as it dissipates 32.0% and 4.5% less than TB-A and TB-B, respectively.

5 A CROSS-LAYER FRAMEWORK FOR EXPLORING APPROXIMATE COMPUTING AND TIMING-SPECULATIVE HARDWARE DESIGN

Current AxC and TS hardware design evaluation methods are agnostic of the impact of the errors in the joint algorithm-accelerators' dynamic and flow-dependency aspects. To overcome this issue, this thesis introduces a seven-step framework to accurately integrate and evaluate the impact of both AxC and TS hardware design. The holistic framework offers the emulation of the hardware accelerators at the gate level to assess the ultimate impact at the application level. Therefore, in addition to logic errors, the framework also embraces errors induced by timing degradation from the physics underlying the hardware accelerator up to the algorithm and application levels.

The following section presents the framework proposal, which allows AxC and TS investigations in different layers across the computing stack (i.e., device physics, circuit, architecture, up to its algorithm interactions). The succeeding sections of this thesis demonstrate the virtue of the framework proposal employing the following case studies:
(**1**) Investigating logic errors using approximate adders (in Section 5.3).
(**2**) Investigating timing errors induced by temperature and aging effects (in Section 5.4).
(**3**) Investigating timing errors induced by VOS (in Section 5.5).

5.1 A Seven-Step Cross-Layer Framework

The framework proposal relies on replacing software functionalities to be hardware-accelerated by their gate-level implementation considering the timing violations modeling and linking them with the application memory and shared libraries employing the DPI-C SystemVerilog (SV) environment. Note that our framework captures the timing errors considering the data-dependency, data-correlation in time, and all the switching activity due to the fully gate-level simulation during the joint hardware accelerator-algorithm online operation application runtime. Fig. 5.1 shows the cross-layer flow implemented in this work. The framework is divided into seven steps as follows:

– Step 1: Performs the cell libraries' characterization with the targeted technology process's design kit and the spice-level netlist with the voltage scaling from the nominal to each point to be evaluated.

– **Step 2**: Adapts the TCTs, for flip-flops (FFs) timing violations modeling into the behavioral Verilog libraries.

– **Step 3**: Performs the logic synthesis to obtain a gate-level netlist of the accelerator block described in RTL, considering the fresh circuit at a nominal voltage (i.e., without timing degradation effects) cell library.

– **Step 4**: Runs the STA considering the cell libraries at each specific case (voltage, temperature and aging). Timing guardbands values are extracted from the reports, and their narrowing is calculated.

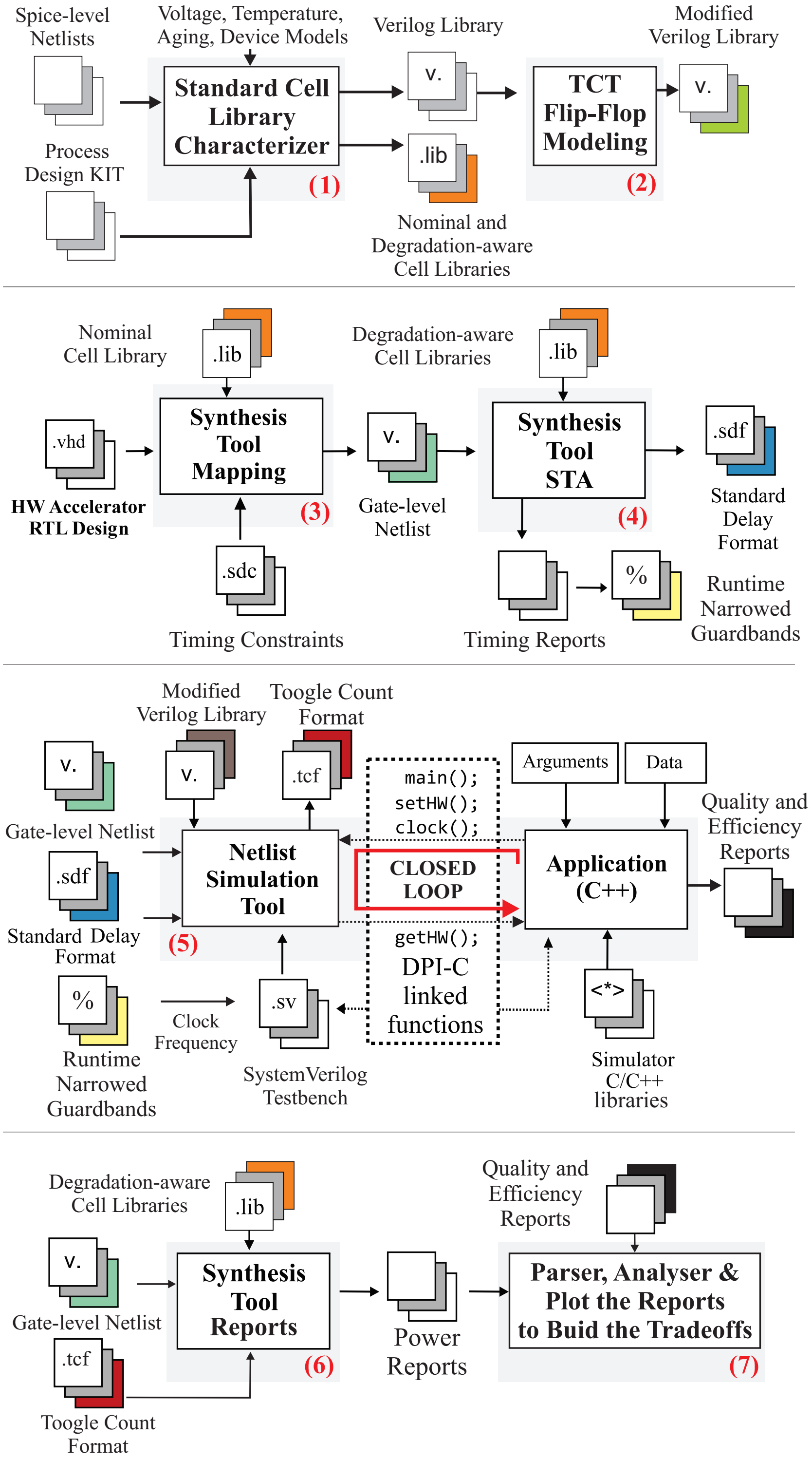
– **Step 5**: Performs the cross-layer calling the netlist logic simulator considering the timing annotation of the STA given by the standard delay format (SDF). The simulation runs at the clock frequency considering the narrowed timing guardbands at runtime. In this step, the simulator integrates the netlist and the application by exchanging inputs and outputs through specific function calls. At this point, the application has the cycle-accurate values from the hardware accelerator under analysis. The outputs of the simulation are the quality and efficiency reports about the application and the toggle count format (TCF) stimuli file.

– **Step 6**: Runs the synthesis tool once again to recalculate the power dissipation reports using realistic stimuli. This step reuses the previously generated gate-level netlist as input (i.e., without performing a logic synthesis again). We employ the TCF stimuli file generated in step (5) and the cell libraries generated in step (1).

– **Step 7**: Evaluates the tradeoff between the application performance versus the energy efficiency, respectively employing the information extracted in steps (3) and (4).

The flow employs an SV model that connects the hardware accelerator netlist (Verilog) to a C++ application using a direct programming interface (DPI), as described in (PAIM et al., 2020), and now including timing impacts. The DPI allows the co-simulation by direct inter-language function calls between the SV and the C/C++ descriptions. It is quintessential that the code is position-independent (fPIC flag) to allow shared access between the simulator and the application. Therefore, DPI allows a direct link between different domains through tasks, functions, arguments, and returns. Commercial tools (as used herein) embed specific header files that are required by the C/C++ application. All linked functions must be defined in both SV and C/C++ with the same signature. The functions can either be of import or export types. The former is implemented in C/C++ and called from the SV, while the latter is described in SV and called from C/C++ functions. At the same time, the SV file instantiates a timed gate-level design under test with the hardware accelerator.

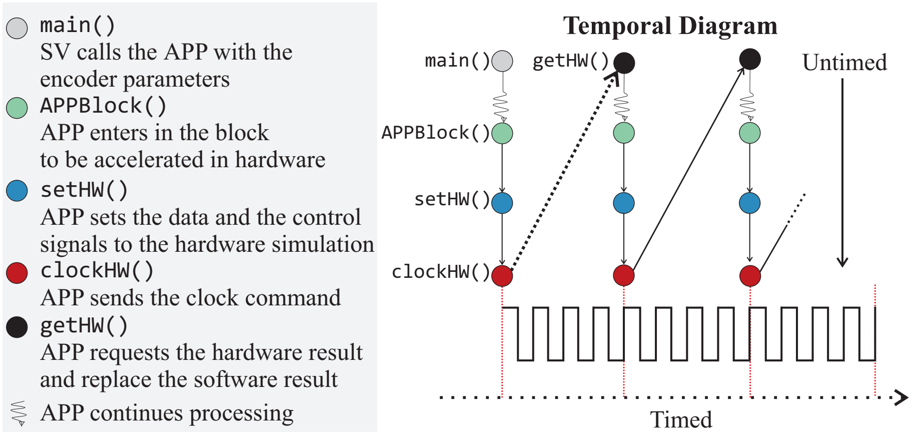
Figure 5.1: The seven-step cross-layer voltage over-scaling flow, where the shadowed region is respective red-colored (n) step.



Source: The Author.

Fig. 5.2 shows the algorithm-accelerator closed-loop flow and its temporal scheduling. Considering the connection between SV and an arbitrary application, we need to implement at least five shared functions to control the hardware accelerator and algorithm: `main()`, `setHW()`, `clockHW()`, and `getHW()`. The testbench in SV manages the entire process as it calls the `main()` function of the application, passing the suitable arguments, as well as instantiating the gate-level hardware accelerator. Assume that the `APP-Block()` function is a generic application compute block whose functionality will be accelerated by the hardware.

Figure 5.2: Temporal diagram between the hardware accelerator within application (APP) execution in runtime (PAIM et al., 2021a).


main()
SV calls the APP with the encoder parameters
APPBlock()
APP enters in the block to be accelerated in hardware
setHW()
APP sets the data and the control signals to the hardware simulation
clockHW()
APP sends the clock command
getHW()
APP requests the hardware result and replace the software result
APP continues processing
Temporal Diagram
main()
getHW()
Untimed
APPBlock()
setHW()
clockHW()
Timed

Source: The Author.

Closed-loop: At each time that `APP-Block()` (i.e. the accelerator) is called within the algorithm, we call `setHW()` untimed function to send the values of both inputs and control of the hardware accelerator to the gate-level simulation. The `clockHW()` function is responsible for timing the simulation by controlling the clock operation. Finally, `getHW()` requests the hardware accelerator results from the simulator and feeds it back to the algorithm-level. This closed-loop simulation process can replace the original software algorithm dynamically with the actual accelerator response that contains logic/timing errors. Ultimately, we can evaluate the algorithm runtime behavior under these conditions through different metrics like the runtime behavior and the quality of the results according to the application target.

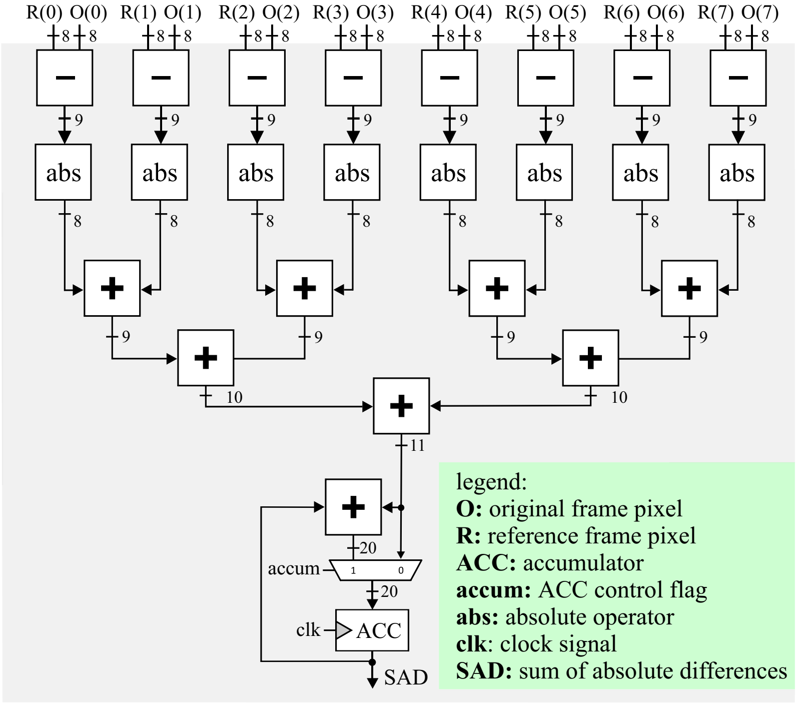
Timing violations occur when the flip-flop (FF) data input is not stable during the setup and hold time windows around the positive clock edge. When the voltage decreases, the hardware accelerator slows down, and the delay increase in the paths. Timing check tasks (TCT) are mechanisms present in the Verilog library primitives to support the gate-level simulations to report the FF timing violations. When setup and hold timing violations occur, the gate-level simulations usually generate '$\times$' (indeterminacy) in the output when it is specified in the TCT. The '$\times$' result case means that the simulation reported a violation and decided to set the FF output value to unknown. Prior work (AMROUCH et al., 2019) propagates '$\times$' value to the output of the open-loop application as a fault. However, in applications with a closed-loop between the hardware accelerators and software application, the '$\times$' state must be replaced by a binary value since it will be propagated and accumulated in the following states of the application. Hence, during the simulations, special provisions must be considered to capture the setup and hold violations and overwrite the '$\times$' states. In our work, we modify the TCT to implement an '$\times$' substitution by either the FF current input (i.e., the FF latches the new value: $Q \Leftarrow D$) or the FF previous output (i.e., the FF maintains its output value: $Q \Leftarrow Q$) (ZERVAKIS et al., 2018) with a uniform probability. It is essential to allow the closed-loop algorithms gate-level simulations and the error accumulation. In the next section, we present the case study implemented to investigate the timing errors employing the cross-layer flow. Due to the intrinsic SystemVerilog (SV) stability when generating uniform pseudo-random values, the pseudo-random values generated are the same when repeating the simulation with equal inputs. This gives stability to our framework, where the one test case can be executed many times, generating the same output result.

5.2 Application Case Study: SAD Accelerator into HEVC Encoder

To demonstrate the virtue of the framework proposal on the AxC and TS hardware design into closed-loop applications, we purposefully chose the SAD hardware accelerator and the fast motion estimation algorithms of the HEVC application as a case study due to its dynamic behavior. Thus, we can evaluate the effectiveness of the implemented method in an industrial-strength accelerator-algorithm co-design. In the following sections, this thesis evaluates as a case study the SAD accelerator in the video encoder employing the cross-layer method presented in the previous section. For our analysis, we also consider the SAD-8 accelerator since it is the most time-consuming and compute-intensive part of the HEVC is the motion estimation (SILVEIRA et al., 2017) that is mainly based on the sum of absolute differences computation.

Fig. 5.3 shows the SAD hardware architecture block diagram. It is composed of eight subtractors to subtract eight samples of the original block (O in Fig. 5.3) with eight samples of the reference block (R in Fig. 5.3). This operation generates the remaining samples, and eight registers store them. The absolute operators receive the signed residues as inputs and perform the absolute operation. An adder tree accumulates the eight absolute values and generates a partial SAD value. At each clock cycle, a register stores a partial SAD value.

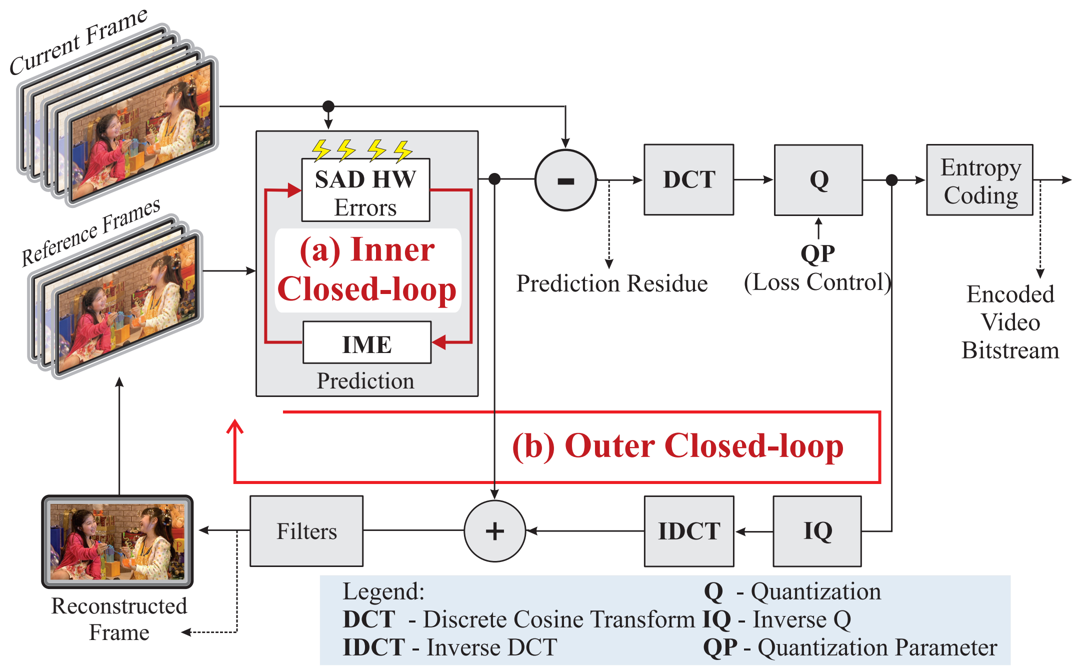
Figure 5.3: SAD hardware accelerator architecture with parallelism of eight comparisons per clock cycle (Eq. 2.1) .



Source: Adapted from (SILVEIRA et al., 2017).

Fig. 5.4 illustrates the encoding closed-loops in the current state-of-the-art hybrid encoders, such as HEVC(SULLIVAN et al., 2012). The video encoder application has more than one dynamic aspect. Fast BMA algorithms are dependent on the current SAD value to decide the search direction and the next steps. Therefore, there is an inner closed-loop between the SAD accelerator and the IME algorithm, Fig. 5.4-a. The outer closed-loop (Fig. 5.4-b) needs to decide the current block partitioning to build the reconstructed frame with the loss included to be used as a reference to the next frames in the prediction. Any error in the SAD hardware accelerator will modify the encoding process's decisions and partitions. Therefore, if the gate-level simulation is made in open-loop outside of the encoder loops, the ultimate result is unpredictable. Open-loop error evaluation in standalone hardware implementations produces an encoder reference mismatch (i.e., drift coding error) and different IME decisions and partitioning. The framework proposal solves this problem, enabling the algorithm-accelerator investigation to consider the logic and timing errors from AxC and TS hardware design.

Figure 5.4: HEVC encoding dynamic behavior (a) inner closed-loop between SAD hardware accelerator and the IME algorithm, (b) outer closed-loop due to the frame reconstruction to be used as a reference frame.



Source: The Author.

5.3 Case Study I: Approximate Adders Design Space Exploration

The approximate computing paradigm (HAN; ORSHANSKY, 2013) emerged as an alternative to improve the energy efficiency by considering that applications with intensive computational processing, such as digital signal, image, and video processing, may use lower-than-standard arithmetic accuracy to provide an acceptable level of information quality. For instance, if the brightness level in a few image pixels is not accurate, the regular viewer might not notice it. In this case, the accuracy might be a tradeoff with power savings. In (AGRAWAL et al., 2016; MITTAL, 2016; SHAFIQUE et al., 2016; XU; MYTKOWICZ; KIM, 2016; BOSIO; MENARD; SENTIEYS, 2018; ZERVAKIS et al., 2019) one can find a survey on approximate computing techniques relevant to this work.

Adder circuits are key logic blocks in hardware accelerators in many error-tolerant applications (GUPTA et al., 2013; SOARES; COSTA; BAMPI, 2016; DUTT; NANDI; TRIVEDI, 2017; ISHIDA; SATO; UKEZONO, 2018; PASHAEIFAR et al., 2019; SOARES et al., 2019), so the use of approximate adders (AAs) is a very appealing alternative for low-power implementations. Notably, the use of approximate arithmetic computing in the coding blocks of the HEVC video standard appears as a promising solution for reducing power dissipation while keeping high-level perceptual information. However, this low-power approach is not widely used in commercial environments given the lack of a well-established evaluation methodology of the effects of approximate arithmetic in the final product. Therefore, this work focuses on employ the framework for accurately build the power versus coding efficiency tradeoff. The framework can build the design space exploration (DSE) by capturing the logic errors in the result of the approximate gate-level circuits in the compute-intensive blocks.

This section demonstrates the virtue of the cross-layer framework to explore approximate adders in a hardware accelerator, taking the entire HEVC encoding process into account. The method supports the DSE of the relevant hardware block and exposes the approximation effects at the visual quality level and the encoder's compression efficiency. The following subsection presents the related work on the DSE of approximate adders into the video encoders.

5.3.1 Related Work

Compute-intensive tasks in video coding, like filtering and the Motion Estimation (ME) unit, use additions very intensively – usually, tens of instances of adders may be present in its parallel hardware implementation. Using AAs in these accelerators is a convenient approach for energy efficiency in digital CMOS design. The work in (ALBICOCCO et al., 2012) shows more straightforward ways to perform addition in an imprecise form, with the objective of high computational performance with smaller area and energy consumption, at the cost of errors that may or may not be tolerable by the target application.

However, most of the literature review evaluates just the errors in standalone arithmetic operations, without a final impact evaluation on the application. As an example, the work in (WU et al., 2019) calculated the error rate and the probability distribution using C++ and applied them to a set of block-based AA to study their performance. The work in (PRABAKARAN et al., 2018; PRABAKARAN et al., 2019) performs an in-depth error analysis using Matlab$^{\text{TM}}$ level behavioral models of the designed adders to understand the output quality of the adder designs, using metrics like Mean Absolute Error (MAE) and error rate (ER) and non-exact probabilistic approaches. It is also proposed a new generic methodology to design AA components based on the target FPGA. The functional models of both accurate and AAs are implemented in Matlab$^{\text{TM}}$ in (LIU; HAN; LOMBARDI, 2015). A set of one million random input combinations are used to find the MAE and ER values for some images.

Evaluating the effect of approximate blocks (like AA) on the entire application system is not trivial, mainly when the application comprises algorithmic dependencies like a current state depending on the data outcomes from the previous clock cycle, as the dynamics of a closed-loop. Video encoding is a class of applications with a high degree of complexity for both the development and verification of the entire system. Moreover, video coding has a dynamic behavior, so an error in the current state may modify all the decisions and actions on the subsequent steps of the encoding process. In this aspect, the SAD accelerator block is an excellent case study for exploring a large set of approximate adders since it is a compute-intensive block highly used in video coding.

The HEVC video encoder incorporates the SAD metric structure in its VLSI implementation, a tree of adders that is highly suitable for approximate circuits. In this context, a SAD architecture is presented in (GUPTA et al., 2013) using a tree of AAs. The approximations in the adders used the same parameters for all of them (i.e., a uniform approximation). Results indicated that the average PSNR ranged from 35 dB up to 37.5 dB for all the adders, being approximated up to four least-significant bits (LSBs). The work in (SHAFIQUE et al., 2015) proposed the generic accuracy configurable adder (GeAr) model and architectural design, error recovery scheme, and error probability model.

Considering the importance of the SAD in the HEVC video coding, existing works have used analytical models of the AAs for reducing both the programming efforts for evaluating the overall effect of errors on the encoding quality. The work in (SHAFIQUE et al., 2016) proposes a cross-layer methodology to assess both power and quality. An equivalent behavioral model (in C or Matlab$^{\text{TM}}$) was developed for output quality evaluation in application scenarios. Selected implementations were also used for error probability analysis. Authors present different variants of the approximate accelerator for the SAD block used in the ME process.

The work in (EL-HAROUNI et al., 2017) performs the error analysis using the number of error cases, maximum error magnitude, and occurrences of maximum error cases. For a complete application evaluation, the equivalent behavior models of the approximate accelerators were built (in C, C++, and Matlab$^{\text{TM}}$), and integrated into an open-source optimized HEVC video coding implementation called x265. The quality evaluation, regarding MV difference (MVD), SAD value, bit rate, and video quality (PSNR).

An architectural exploration in a variable block size ME (VBSME) architecture using the imprecise Lower-Part-Or Adder (LOA) is presented in (PORTO et al., 2017; PORTO et al., 2019). The adders were employed in the SAD architecture to reduce the energy consumption while introducing a minimum impact on coding efficiency. The considered approximate operators were described in C++ to replace part of the source code in the HM 16.12 HEVC reference software (HEVC..., 2016). The latter enables the evaluation of the impact of the proposed computing imprecision on the resulting coding efficiency. The authors in (PORTO et al., 2019) performed the tradeoff analysis considering just one adder configuration. The reported results showed the power dissipation vs. coding efficiency for only one AA case with LOA with five approximation bits uniformly applied to all adders in the tree, lacking a DSE to search the optimal case for a given preset loss target.

Five different AAs such as LOA, ACA, ACAA (Accuracy Configurable AA), ETA-I (Error-Tolerant Adder I), and SCSA (Speculative Carry Select Adder) were used in three different places inside the SAD architecture in (PALTRINIERI et al., 2018). The analysis of the results considered both error and power savings. The SAD architecture was described in a C++ model based on the SAD accelerator described in (SELVO et al., 2018). This model reproduces the bit-accurate behavior of the architecture. After evaluating in software the various scenarios of approximations, the modified adders were then described in VHDL in the SAD architecture and synthesized on a 65 nm standard cell technology.

The work in (ALAN; HENKEL, 2018) proposed a logic synthesis strategy to allow graceful timing errors on datapath circuits operating at over-clock frequencies. The key idea is that the timing requirements can be relaxed, allowing timing errors. However, the results were based on circuit-level analysis only, considering the SAD datapath's output error without a methodology capable of evaluating timing errors in the application.

In the previous works, the evaluation of approximate operators in the target application is done by developing these arithmetic operators' models, evaluating the impact for a few approximate architectures and configurations (in Table 5.1 for related works feature differences). Our work proposes a co-simulation method that uses a cross-layer strategy to simulate the gate-level within the existing application software model. To the best of our knowledge, none of the works of the literature evaluated the impact on the entire video coding standard (power vs. coding efficiency) on the fly when using a gate-level approximate SAD block, as this work herein proposes. The proposed method directly integrates video coding software with industry-standard logic circuit simulators. The co-simulation method allows the video coding system designer to precisely estimate the approximate hardware optimization impact by including the gate-level circuit directly into the application. Our method eliminates the need to create additional models or translate the approximate arithmetic circuit into a programming language for each configuration level and is suitable for timing-error evaluation.

Table 5.1 summarizes the different approaches by ten other works in the context of approximated blocks for video and our contributions with a co-simulation modeling and assessment of the actual impacts of AAs for the SAD metric. Our approach is non-statistical (A column) and cycle-accurate, while being unique in its cross-level gate-level co-simulation to assess coding efficiency (G column).

Table 5.1: Summary of Related Work.

	Related Work Summary Results								
Related Work	A	B	C	D	E	F	G	H	I
(ALBICOCCO et al., 2012)	–	–	✓	–	–	–	–	2	6
(GUPTA et al., 2013)	–	–	✓	✓	–	–	–	6	24
(SHAFIQUE et al., 2015)	–	–	✓	–	–	–	–	5	13
(LIU; HAN; LOMBARDI, 2015)	–	–	✓	–	–	–	–	3	21
(SHAFIQUE et al., 2016)	–	–	✓	✓	–	–	–	3	9
(PORTO et al., 2017)	✓	–	–	–	–	–	–	1	3
(EL-HAROUNI et al., 2017)	–	–	✓	✓	–	–	–	3	9
(PALTRINIERI et al., 2018)	–	–	✓	✓	–	–	–	5	15
(ALAN; HENKEL, 2018)	✓	–	–	✓	–	–	–	0	4
(PRABAKARAN et al., 2018)	–	–	–	–	–	–	–	8	16
(PRABAKARAN et al., 2019)	–	–	–	–	–	–	–	8	16
(PORTO et al., 2019)	✓	–	–	–	–	–	–	1	1
(WU et al., 2019)	–	–	–	–	–	–	–	3	6
This work	✓	✓	✓	✓	✓	✓	✓	13	3,074

(A) Non-statistical model for the AAs,

(B) Video sequence input for realistic gate-level power analysis,

(C) Approximate SAD analysis using more than one AA,

(D) Different levels of approximation in SAD,

(E) Cross-layer from gate-level to application for approximate circuits evaluation,

(F) SAD with variations of copy AAs,

(G) Power Dissipation vs. compression efficiency tradeoff results for the entire video coding from gate-level approximate SAD,

(H) # of different approximate circuits architectures evaluated in the tradeoffs,

(I) # of approximate circuits configurations evaluated in the tradeoffs.

Observing the pipelined SAD architecture structure (Fig. 5.5), the adder tree's intermediate values are composed of three levels of additions after the absolute blocks. The number of bits of the operands in the SAD tree architecture is 8, 9, and 10 for the first, second, and third levels. Hence, for the approximate SAD, we clustered each level and applied a complete search for approximations with the K parameter varying from 1 until $N - 1$ (where N represents the operand bit-width).

The K_1, K_2, and K_3 parameters are highlighted in Fig. 5.5 for the first, the second, and the third level, respectively. We used the set of 13 AA discussed in the previous subsection at each of the DSE levels. The variations in the parameter K defined the limits of approximations that can be applied to maintain the video encoding quality.

Table 5.2 summarizes all the K approximations explored on the DSE of this work. For truncation, copy, ETA-I, and LOA AA architectures, the higher the approximation in the K parameter, the less power is dissipated in the SAD architecture. On the other hand, for the adders block-based adders like ETA-II, ETA-IIM, and ACA, higher K is higher precision and higher power dissipation. Based on the circuit considerations and the DSE performed, we restrict the K search space with this criteria or bounds:

Truncation, copy: K_1, K_2, K_3 ranging from 1 to 6. Above K = 5, the coding efficiency is seriously degraded, with a more than 2% increase in BD-BR.

ETA-II and ETA-IIM: K_1, K_2, K_3 range from 3 to 6 in the DSE. K starts with a value equal to 3, given the size of the internal CLA adder. For K higher than 6, the size of the CLA block is inefficient.

ACA: K_1, K_2, K_3 ranging from 3 to 7, 8, 9. The ACA with block size less than 3 is very imprecise since the ACA incurs a high magnitude error for small blocks.

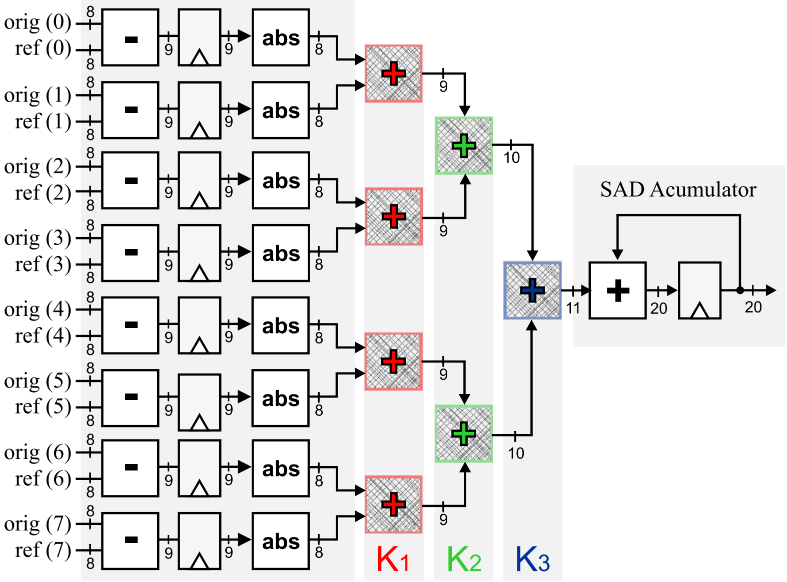
ETA-I, LOA: K_1, K_2, K_3 ranging from 1 to 7, 8, 9. In the same manner as in truncation and copy, as these adders are not block-based, the range of all K values can be covered starting from K = 1. ETA-I and LOA adders kept coding efficiency with higher K values.

Table 5.2: Approximate SAD K configurations summary.

AA Architectures	K_1	K_2	K_3	Versions Count
2×Truncation[1], 6×Copy[2]	1–6	1–6	1–6	1728
ETA-I, LOA	1–7	1–8	1–9	1008
ETA-II, ETA-IIM	3–6	3–6	3–6	128
ACA	3–7	3–8	3–9	210
# of Total configurations				3074

[1] Trunc$_0$, Trunc$_1$; [2] CopyA$_{\mathrm{AND}}$, CopyB$_{\mathrm{AND}}$, CopyAB$_{\mathrm{AND}}$, CopyBA$_{\mathrm{AND}}$,. CopyA$_{\mathrm{Copy}}$, CopyB$_{\mathrm{Copy}}$.

Figure 5.5: The K levels of SAD where the approximate adders were applied.



Source: The Author.

5.3.2 Impact of the Approximate SAD on HEVC

More than 3,000 SAD hardware architectures were described in VHDL using 13 different configurations of AAs scripted on Matlab^{TM} language to generate the different K approximation levels. The precise part of the AAs was designed utilizing the adder selected by the synthesis tool (for the plus operator in VHDL) except for the ETA-II adder that employs specific blocks of RCA and CLA adder architectures. The precise baseline adder used in all comparisons was described also using the precise adder selected by the synthesis tool.

The Cadence Genus^{TM} tool was employed to perform the RTL to gate-level netlist synthesis, STA, and the QoR reports, mapping to the low-power 65nm ST standard cell industrial library at 1.0V voltage supply. The synthesis target clock is set to 600 MHz (with zero slack value). According to (SILVEIRA et al., 2017) that considers a realistic case evaluation, using this target frequency, a SAD architecture with a parallelism of 8 compared pixels per cycle can processes videos of 1920×1080p resolution at 30 frames per second in real-time. The Cadence Incisive^{TM} was used to perform the gate-level netlist simulations considering the delays, reporting for all video sequences a realistic TCF file to evaluate the power dissipation analysis in Cadence Genus^{TM}.

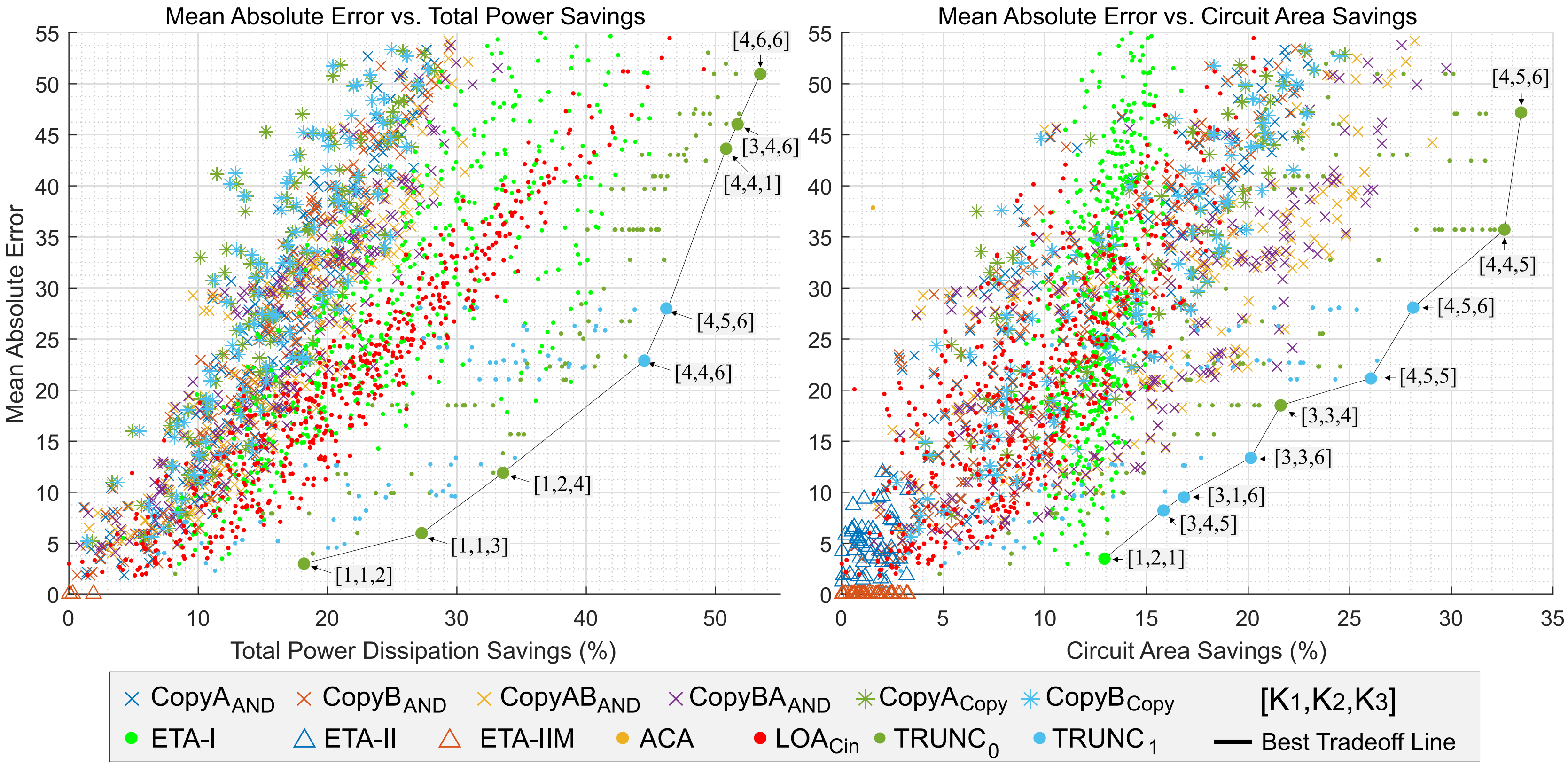
The impact of the approximation on SAD trees can be assessed with either a circuit- or an application-level method (see Fig. 5.6 compared to Figs. 5.7 and 5.8 respectively). The circuit-level analysis is based on the accelerator output error assessment strategy and cannot provide the coding efficiency of the impact into the final application. The circuit-level analysis was performed with default power dissipation results from the synthesis tool, the common methodology utilized in most related works to evaluate approximate circuits. The approach simulates the circuit separately, feeding the architecture with inputs extracted in batch from x265 and reading the outputs to compare the error. The circuit-level analysis cannot provide the impact of the error on the final application.

The application-level analysis results were based on the proposed cross-layer gate-level-to-application co-simulation method that, for the first time, enables the evaluation of the actual impact of gate-level approximate circuits on the entire HEVC encoder operation. The analysis of this thesis is targeting low-power mobile applications. Therefore, our simulations consider the x265 encoder implementation, an optimized HEVC implementation, in its super-fast preset.

5.3.3 Circuit-level Results

The main goal of the circuit-level tests shown in Fig. 5.6 is to demonstrate that for error-tolerant applications, such as video coding, the magnitude of the errors may not be directly correlated with its performance inside the complete application at hand. The circuit-level simulation of the SAD gets the errors in the output of the SAD tree. MAE metric was used to evaluate the error in the output of the SAD architecture, according to equation (3.9). In the equations, $Exact$ represents the output using the precise adder, and $Approx$ means the output using the AAs. For this purpose, we perform the MAE on average using four Quantization Parameters (QPs) (equal to 22, 27, 32, 37) for 100,000 samples using the SAD pattern encoding the RaceHorses (RH) 240×480p video sequence at 30 frames per second (fps). These QPs configurations were defined in the HEVC common test conditions (CTCs) document (BOSSEN et al., 2013).

Figure 5.6: DSE results for MAE versus power and area savings with the set of AAs (Tab. 5.2) in the SAD architecture.



Source: The Author.

The circuit-level errors of the SAD for both: i) MAE vs. power dissipation savings with and ii) MAE vs. circuit area savings are almost linear with the approximations, as can be seen in Fig. 5.6. This occurs because the errors are only evaluated from the input to the output of the SAD adder tree, with no consideration to the dynamic aspects. Notably, in the circuit-level analysis, the SAD with the truncation AAs demonstrate the optimal tradeoff results on all the wide ranges.

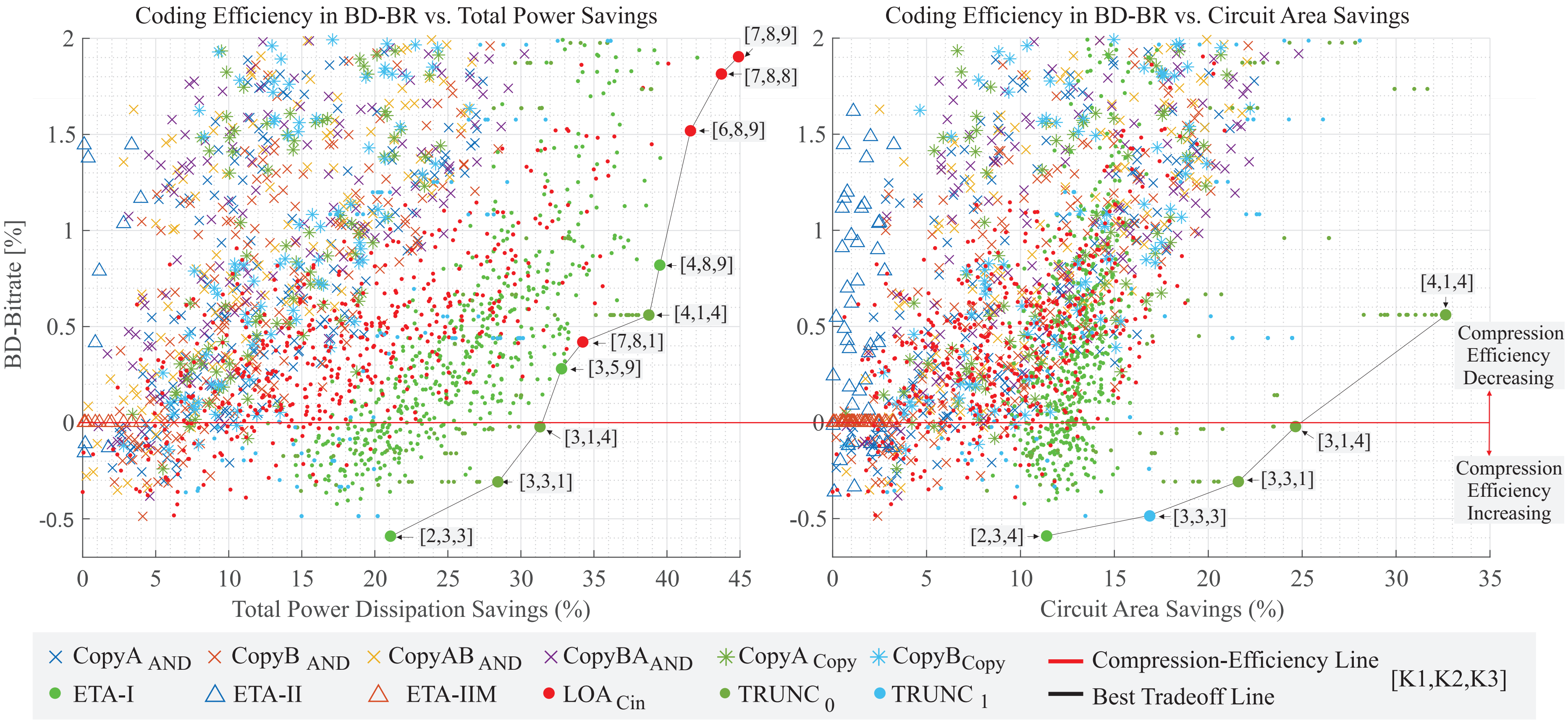
As expected, the SAD with ETA-II and ETA-IIM are among the worst results for both circuit area and power dissipation positive savings with lower MAE errors. This occurs because both ETA-II and ETA-IIM need both CLA carry chain and RCA adders to build the sum. The power and area savings of ETA-II and ETA-IIM are caused by the shortening of the CPD, which reduces the required drive strength of the transistors in the synthesis. ACA adder does not provide power- or area- savings. Excepting the case with K_1= 3, K_2= 4, and K_3= 3 with only 1.53% of area savings and high error, that results in a worse tradeoff. This occurs because the ACA AA is divided into K-bit overlapping blocks to improve the computational performance, obtained at the cost of higher area and power. This aspect contributes to the ACA not being classified as a power-oriented adder.

We show in the following subsection that the same AA circuits under realistic application-level analysis demonstrate a different power versus compression best tradeoff for two reasons: (a) the quality was not measured in terms of the error but for coding efficiency, which is the more important metric for video encoding. (b) the power dissipation results are measured using realistic stimuli, real delays in the netlist, and realistic video encoding scenario using the proposed method in-the-loop.

5.3.4 Application-level Results

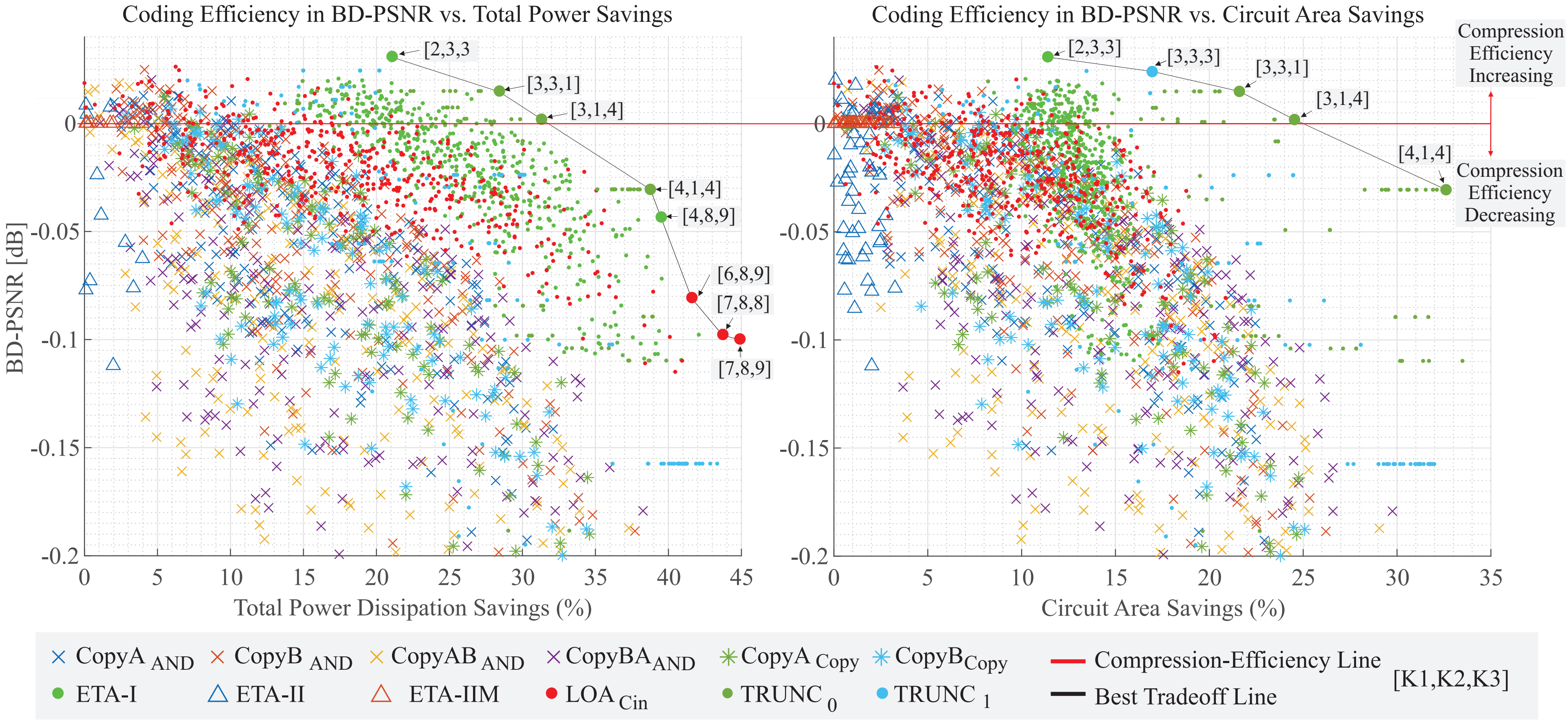
The application-level analysis evaluated by the co-simulation represents the real impact of the SAD approximations into the entire encoder, capturing all the circuit behavior. The application-level analysis presents results for coding efficiency versus realistic power dissipation and circuit area. The proposed gate-level-to-application co-simulation method, using the approximate SAD block, was exercised with the same four QPs (22, 27, 32, 37) and one second of the RH 416×240 video sequence at 30 fps.

Figure 5.7: DSE results for BD-BR coding efficiency versus power and area savings with the set of AAs (Tab. 5.2) in the SAD architecture.



Source: The Author.

Figure 5.8: DSE results for BD-PSNR coding efficiency versus power and area savings with the set of AAs (Tab. 5.2) in the SAD architecture.



Source: The Author.

Figs. 5.7 and 5.8 shows the results for the DSE AA impact on HEVC coding efficiency for BD-BR and BD-PSNR, respectively, vs. power dissipation and circuit area savings. The comparison uses the BD-BR measurement method. Negative values inform how much lower the bitrate is (a positive outcome), and positive values mean a bitrate increase for the same PSNR quality.

The SAD with all sets of AAs causes a coding efficiency increase in a moderate region of power gains (up to 32%). It is, in fact, possible because the BMA is a sub-optimal algorithm, and SAD errors can eventually lead to better MVs. The ACA presents the worst results among the AAs, without savings, and did not appear in the Figs. 5.7-5.8. Such was expected since, depending on the conventional adder topology, this adder may incur a substantial circuit area and power dissipation cost.

The circuit-level analysis shows that the SAD error with the ETA-I adder increases asymptotically with only the truncations AAs on the best trade-ff. The application-level analysis shows that the best tradeoff between power and coding efficiency includes the ETA-I and LOA AAs. Application-level analysis using the ETA-I results demonstrates that this adder can show coding efficiency gains simultaneously that it achieves up to 21% power savings.

At the application level, the ETA-II demonstrates much worse error results when compared with the circuit-level analysis. It occurs because the ETA-II behavior presents an infrequent high-magnitude error, and this error behavior is catastrophic for SAD. An infrequent high magnitude error behavior misguides the BMA that compares different candidate blocks and can directly cancel the high-quality MV candidates, changing the search direction when the high-magnitude errors occur. Another observation about the circuit-level analysis's imprecision is that the levels tuning on the best tradeoff between both analyses are misleading. Neither truncation adder using K>5 presents acceptable approximations in the realistic application-level analysis that we introduced.

Table 5.3: Optimal tradeoff AA from Figs. 5.7 and 5.8 for BD-BR and BD-PSNR versus the synthesis results.

				BD-BR	BD-PSNR	Static Power	Dynamic Power	Total Power	Power Savings[2,3]	Circuit Area	Area Savings[2]
AA[1]	K1	K2	K3	(%)	(dB)	(uW)	(uW)	(uW)	(%)	(μm)	(%)
ETA-I	2	3	3	-0.5903	+0.03096	1.83	2557.1	2558.9	21.07	2535	11.39
TRUNC$_1$	3	3	3	-0.4862	+0.02441	1.72	2585.7	2587.4	20.19	2378	16.88
TRUNC$_0$	3	3	1	-0.3077	+0.01495	1.41	2319.2	2320.6	28.42	2243	21.60
TRUNC$_0$	3	1	4	-0.0212	+0.00192	1.24	2225.9	2227.1	31.31	2159	24.54
ETA-I	3	5	9	+0.3195	-0.01672	1.60	2170.2	2171.8	33.01	2424	15.27
LOA	7	8	1	+0.4201	-0.02241	1.71	2130.5	2132.2	34.23	2373	17.06
TRUNC$_0$	4	1	4	+0.5602	-0.03053	1.07	1986.9	1987.9	38.69	1928	32.61
ETA-I	4	8	9	+0.8200	-0.04330	1.62	1959.5	1961.1	39.51	2425	15.17
LOA$_{\text{Cin}}$	6	8	9	+1.5190	-0.08057	1.61	1891.6	1893.2	41.61	2297	19.71
LOA$_{\text{Cin}}$	7	8	8	+1.8150	-0.09763	1.60	1822.8	1824.4	43.73	2281	20.27
LOA$_{\text{Cin}}$	7	8	9	+1.9040	-0.09973	1.60	1784.9	1786.5	44.90	2264	20.87

[1] The precise part was described using the adder automatically selected by the synthesis tool. [2] The baseline is the precise adder automatically selected by the synthesis tool.

🟢 Green is a better compression-efficiency. 🔴 Red is a worse compression-efficiency.

The plots in Figs. 5.7-5.8 show the essential parts of the application-level test's results for DSE (providing power savings in a range of 0 to 45%). The red line represents the reference (without losses), while the black line represents the Pareto curve with the best tradeoff between coding efficiency versus area and power savings. The curves also present the values of K_1, K_2, and K_3 for the best AAs in SAD. Table 5.3 summarizes the results of the best tradeoff configurations of the Pareto curve presented in Figs. 5.7-5.8 with numeric values and also including HEVC coding efficiency versus power dissipation results. According to the results, if no compression losses are allowed, the HEVC with the SAD based on truncation adder ($TRUNC_0$) is the best choice, with K_1=3, K_2=1, and K_3=4 values, when considering both BD-BR and BD-PSNR metrics. In this case, the most important accelerator of the HEVC encoder can save more than 31.31% in total power and 24.54% in the circuit area. However, if some coding efficiency losses are allowed (up to 1.9% in BD-BR, and until -0.1 dB in BD-PSNR), the area and power savings can be more expressive. At the maximum loss of 2% in BD-BR and 0.1 dB in BD-PSNR, the HEVC with the SAD based on the LOA adders present higher power dissipation savings.

Notably, the SAD with LOA adder can provide savings in power dissipation of close to 45%. For these substantial gains, the SAD with LOA AA presents values of K_1, K_2, and K_3 equal to 7, 8, and 9, which amount to almost the total size of the operands with approximations. The plots also show that the increase in coding efficiency is allowed in HEVC by the truncation ($TRUNC_0$ and $TRUNC_1$) and ETA-I AAs in SAD when circuit area savings are considered. Therefore, we can obtain area gains without loss in coding efficiency in a range between 15% and 24%. On the other hand, only the HEVC with SAD based on $TRUNC_0$ and ETA-I AA allows an increase in coding efficiency, with benefits for the range of 21% and 31% in power dissipation.

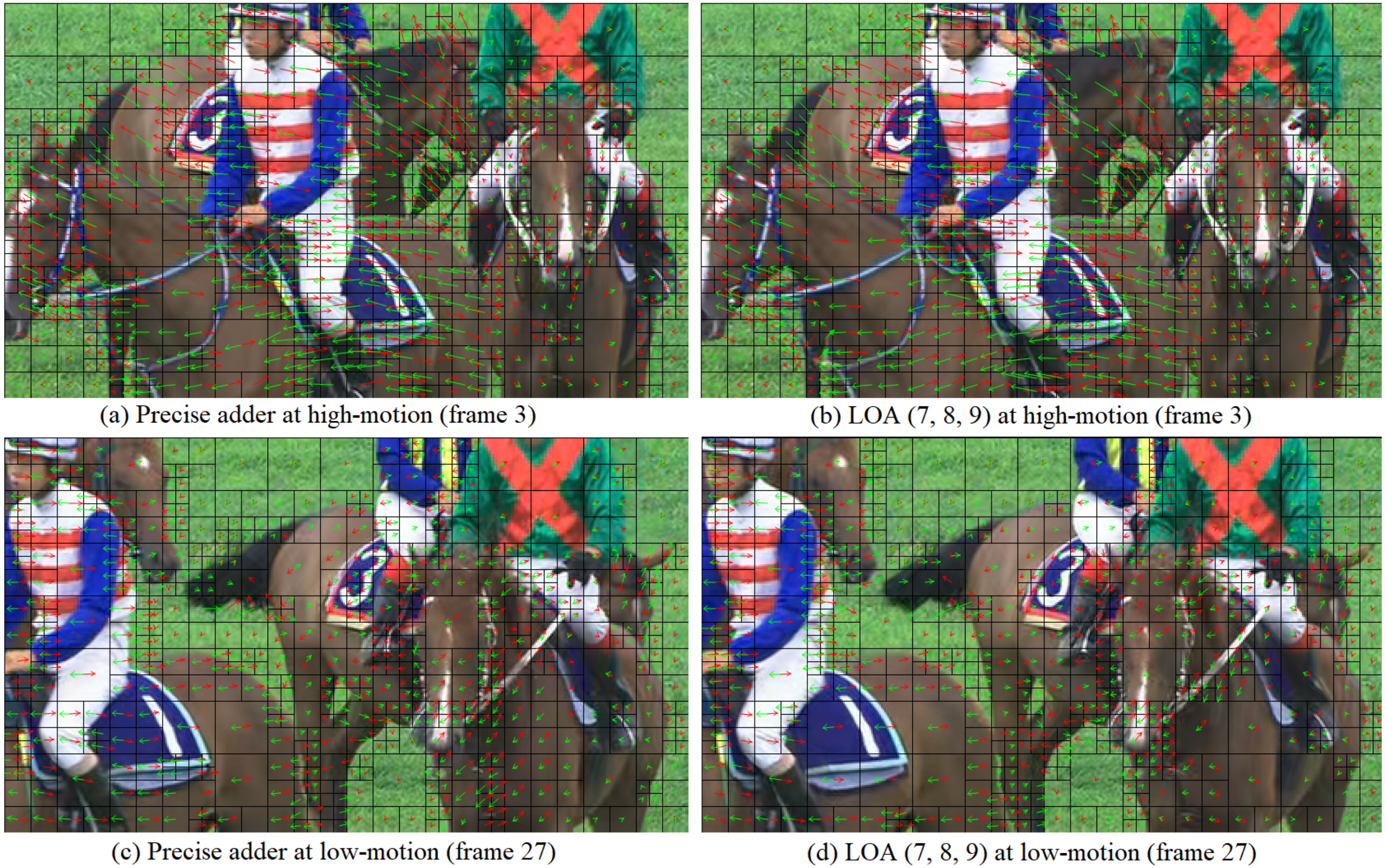
The obtained results show how SAD is a greatly error-resilient block for employing approximations. In fact, as in the SAD, the values of the sum of the absolute differences are accumulated, and, therefore, more variations occur in the least significant bits, where the approximations are applied. As can be seen, by using the LOA AA in the SAD, it is possible to obtain gains of close to 45% power dissipation, with a slight decrease in coding efficiency. It proves the essence of the SAD block to be appropriate to the use of approximations. Another essential aspect being considered is the better gains in area and power obtained in HEVC with the SAD using LOA, ETA-I, and truncation AA (see Table 5.3), which confirms that these AAs are power-oriented circuits. Mainly, the experience with the truncation (by truncating the least significant bits to 1 - $TRUNC_1$) is essential to obtain some benefits in the area with a slight improvement in coding efficiency. On the other hand, the ETA-II, ETA-IIM, and ACA did not provide significant SAD savings regarding the circuit area and power dissipation, which confirms that these AAs are performance-oriented. The copy adder did not enable any significant result in the coding efficiency versus power dissipation and circuit area tradeoff in SAD compared with the other power-oriented explored versions.

5.3.5 Motion Vector Impact Analysis

Fig. 5.9 shows an MV analysis for high- and low-motion videos for both the precise (a,c) and LOA (b,d) to observe the impact of the approximations, using respectively the frames 3 and 27 of the RaceHorses sequence. The MV analysis is performed for the same video and encoder configurations as the other analysis. However, with the QP set to 32. To perform the MV analysis, we choose the LOA K_1=7, K_2=8, and K_3=9 that resulted in the higher power savings accepting up to 1.9% of BD-BR loss (see Table 5.3). The red-colored arrows in Fig. 5.9 represent L0 MVs, and the green-colored are for L1 MVs.

The MV analysis results show that the approximate adder maintains almost the same MV for high and low-motion. For high-motion, there is less impact than low-motion for one's surprise. However, tiny vectors tend to result in lesser magnitude values between the comparisons of SADs, resulting in similar reasonable solutions in the neighborhood of small vectors. Therefore, vectors with a small magnitude (from low-MV) tend to result in the most challenging decision based on a SAD result using an approximate adder. The results of these analyses for low-motion show that the approximate SAD tends to select a higher partitioning for homogeneous regions, and that is acceptable for the encoding, as seen in Fig. 5.9-(c) and (d) on the body of the horses.

Figure 5.9: High- and low-MV analysis to evaluate the impact of AA in the HEVC video encoding: the precise adder versus the LOA with K_1=7, K_2=8, and K_3=9.



(a) Precise adder at high-motion (frame 3)

(b) LOA (7, 8, 9) at high-motion (frame 3)

(c) Precise adder at low-motion (frame 27)

(d) LOA (7, 8, 9) at low-motion (frame 27)

Source: The Author.

5.3.6 Comparisons with the State Of the Art

A direct comparison of our results with the related work is challenging to provide. There are no works that simulate the gate-level circuit dynamically inside the entire HEVC video coding. Three prior work has proposed integrating SAD with AAs into the x265 software model, but with a post-design evaluation with no attention to the dynamic progression of the SAD to the encoder.

The work in (PORTO et al., 2017; PORTO et al., 2019) demonstrated both power and coding efficiency results only for one LOA AA configuration, into SAD, while our work explored herein a more extensive set of AA types and a higher number of bit-level approximations in the SAD architecture. The approximate operator description in (PORTO et al., 2017; PORTO et al., 2019) were described in C++ to replace part of the source code in the HM 16.12 HEVC reference software. The synthesis results indicate power savings from 9.7% to 22.1%, with BD-Rate increasing from 0.6% to 2.5%. Our work shows that by using LOA adder in SAD, it was possible to achieve up to 45% of power savings, with a tolerable loss of almost 1.9% in BD-BR and approximately 0.1 dB in BD-PSNR. The work in (PORTO et al., 2017; PORTO et al., 2019) obtained the power savings with a maximum approximation of 5-bits in LOA adder. This work shows that the exploration in LOA can be more aggressive by approximating it up to 7, 8, and 9 bits with still more power savings achievable for acceptable coding efficiency impacts. Moreover, the work in (PORTO et al., 2017; PORTO et al., 2019) explored just one level of approximation in SAD, while our results take into consideration more than 3,000 configurations for the SAD approximation with a comprehensive DSE.

The work in (EL-HAROUNI et al., 2017) shows SAD results with three AAs, resulting in the best adder as AppxAdd3 (named by the authors), which had the same architecture of our copy^{A}. The cited work only presented bitrate results (kbps) and PSNR, while our work showed a complete analysis of BD-BR and BD-PSNR. Moreover, the cited work uses only the maximum of 6 LSBs for approximation, which, according to the authors, results in a very high increase in the bitrate, which may be unacceptable. Our work showed that with a slight loss in BD-BR and BD-PSNR, gains close to 45% power dissipation and 32% in circuit area are achievable. Finally, the work in (EL-HAROUNI et al., 2017) developed the equivalent behavior models of the approximate accelerators (in C, C++, and Matlab$^{\text{TM}}$), and integrated them into an open-source optimized HEVC implementation called x265.

The prior works employed behavioral and statistical models of the approximate accelerators. Our work integrates the gate-level SAD with the AAs into the x265 model software, which allowed a more realistic tradeoff between results in the area, power, and coding efficiency. After implementation, our method also was the first that permits the designer to enter only with the approximate RTL version without another extra application software modification.

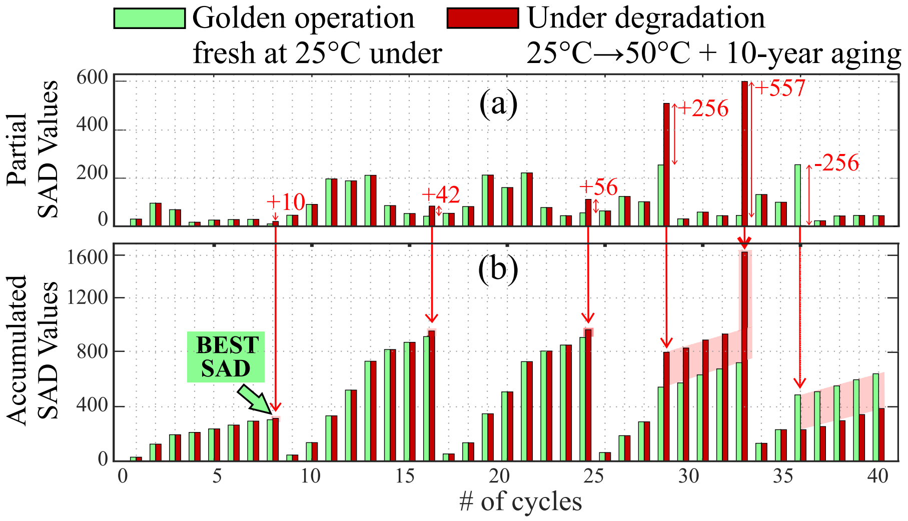
5.3.7 Conclusions of the Section

This section presented for the first case study showing the cross-layer framework that permits the variation and analysis of the gate-level circuit behavior, bringing the gate-level logic design impact up to the target application dynamically. The HEVC video coding is used as a case study to evaluate the coding efficiency when design AA circuits. The sum of absolute differences similarity metric in the HEVC was used to test the cross-layer method since it is one of the most time-consuming operations in the HEVC encoder. It was possible to examine different AAs into the tree of additions of SAD. Circuit- and application-level tests were realized to ensure the better accuracy of the proposed cross-layer co-simulation method. The experiments shown as unrealistic are the only circuit-level results, which fail to account for the application-level behavior of the HEVC video coding in its entirety. As an example, the circuit-level test shows that the error in SAD with ETA-I adder increases asymptotically. However, the application-level test results showed that the HEVC with the SAD based on ETA-I and LOA presents up to 45% of power reduction with a slight increase of only 1.9% in BD-BR, and 0.1 dB in BD-PSNR. If no losses in both BD-BR and BD-PSNR are allowed, the HEVC with the SAD based on Truncation-to-zero provided the best results. Finally, we also presented an MV analysis for high- and low-motion to observe the approximations' impact for the most power-efficient AAs in the DSE. The MV analysis shows a low impact on both the MV decision and partitioning decision.

5.4 Case Study II: Temperature and Aging Effects

To demonstrate how degradation-induced timing errors impact the algorithms, we purposefully chose the SAD hardware accelerator, along with BMA algorithms, as a case study to evaluate the effectiveness of the implemented framework in an industrial-strength accelerator-algorithm co-design. We evaluate the temperature degradation impact on the BMA using the framework presented in the previous section. We assume the baseline temperature to be room temperature of 25°C and explore the impact of temperature rise. We study two higher temperatures of 50°C and 75°C. To explore the effect of transistor aging, we consider a lifetime of 10 years in which a threshold voltage increase of 50mV in transistors due to the BTI is induced. Since the assumed operating temperature range is from 25°C to 75°C, we consider the average temperature of 50°C when studying the effects of transistor aging. Note that our work is not limited to specific temperatures or operation lifetime. Other scenarios can be analogously analyzed. The corresponding degradation-aware libraries are created and then plugged into our framework to investigate how degradation-induced timing errors affect algorithms' runtime behavior.

Figure 5.10: Illustration of SAD degradation-induced errors: (a) Partial SAD value at each single cycle (b) Accumulated 8×8 block SAD values.



Source: The Author.

Fig. 5.10 illustrates the SAD process with degradation-induced timing errors compared with the golden hardware accelerator to demonstrate the SAD resilience in the context of the BMA. Fig. 5.10(a) shows the SAD being calculated in partial chunks (8-pixel SAD chunks), and each of these chunks is accumulated to determine the total SAD for an 8×8 SAD block (Fig. 5.10(b)). Some SAD chunks tend to be impacted by a statistical noise in their values under degradation with the possibility of bit flips, which results in different values compared to the golden model for hardware accelerator operation. However, the best SAD (smallest value) choice can be maintained the same at the end of the execution, as demonstrated in Fig. 5.10.

Table 5.4: Summary of the related work.

	Summary Results								
Related Work	A	B	C	D	E	F	G	H	I
(AMROUCH et al., 2019)	✓	✓	✓	–	–	–	–	✓	–
(HE; GERSTLAUER; ORSHANSKY, 2013)	–	–	–	–	–	–	–	✓	–
(CHENG et al., 2018; VALLERO et al., 2019)	–	–	–	–	✓	–	–	–	–
(ZERVAKIS et al., 2018)	–	–	–	–	–	–	✓	✓	–
(MOGHADDASI; NASAB; KARGAHI, 2020)	–	–	✓	–	–	–	–	–	–
(AFZALI-KUSHA et al., 2020)	–	✓	–	–	✓	–	–	✓	–
(WANG; ROBINSON, 2019)	–	–	–	✓	–	–	–	✓	–
(JIAO et al., 2018)	–	✓	✓	✓	–	–	–	✓	–
This work	✓	✓	✓	✓	✓	✓	✓	✓	✓

(A) Investigates timing errors at gate-level simulations employing degradation-aware standard cells.

(B) Capable of evaluating timing-errors induced by temperature degradation.

(C) Capable of evaluating timing-errors of aging effects.

(D) Enables the investigation of any kind of degradation-induced timing-errors on the joint accelerator-algorithm and its runtime impacts.

(E) Degradation-induced effects modeling considering FinFET devices.

(F) Device modeling fully calibrated within real measurements from industrial technology redprocesses (such as Intel).

(G) Supports both combinational and sequential circuits.

(H) Captures the timing errors considering both data-dependency and data-correlation in time.

(I) Bridges the gap dynamically from gate-level in-the-loop with the application to support the investigation of closed-loop algorithms.

5.4.1 Related Work

Most works that analyze tradeoffs between timing errors and hardware efficiency focused on investigating isolated layers without connecting the hardware accelerators with the algorithm and application layers. A comprehensive survey about this issue is presented in (STANLEY-MARBELL et al., 2020), showing that the most critical challenge is the quest for holistic cross-layer approaches. Our framework advances the state-of-the-art in this direction, allowing accurate investigations on the temperature-induced timing errors within a framework compatible with any accelerators, algorithms, and applications (i.e., embracing closed-loops). Table 5.4 presents the characteristics of several related works found in the literature, comparing them to our work.

Cross-layer reliability frameworks for radiation-induced soft errors investigations are proposed in (VALLERO et al., 2019; CHENG et al., 2018). The cross-layer approaches in (VALLERO et al., 2019; CHENG et al., 2018) are based on modeling faults into the memories without cover the errors into combinational circuits. The frameworks in (VALLERO et al., 2019; CHENG et al., 2018) did not support the investigation of the timing errors induced by temperature and aging degradation effects.

The works in (WANG; ROBINSON, 2019; HE; GERSTLAUER; ORSHANSKY,

2013) investigate the acceptance and techniques to mitigate timing errors induced by voltage reduction at the circuit layer. In (WANG; ROBINSON, 2019), the timing errors are generically evaluated at the output of circuits based on ISCAS85 benchmarks without a specific application target as well as any specific degradation-induced effect. The work in (HE; GERSTLAUER; ORSHANSKY, 2013) presents stand-alone DCT/IDCT hardware accelerators under timing errors showing results only for image compression applications at the circuit layer (i.e., in open-loop). The work in (HE; GERSTLAUER; ORSHANSKY, 2013) cannot present the ultimate impact analysis in terms of compression efficiency due to the limitation of performing the gate-level synthesis (GLS) in an open-loop approach. A direction as future work highlighted in (HE; GERSTLAUER; ORSHANSKY, 2013) is to evaluate the impact of the timing errors underlying hardware accelerators in the full context of the video compression (i.e., within the encoder's closed-loop).

The work in (ZERVAKIS et al., 2018) offers a framework to investigate the acceptance of timing errors in the output of stand-alone hardware accelerators. As in (HE; GERSTLAUER; ORSHANSKY, 2013), the work in (ZERVAKIS et al., 2018) evaluated video accelerators, although they do not assess the ultimate impact of the timing errors at the algorithm and application layers. Despite the limitations of crossing the layers, the central contribution in (ZERVAKIS et al., 2018) is to offer modeling for enabling sequential circuits timing errors evaluation for GLS.

The frameworks in (JIAO et al., 2018; MOGHADDASI; NASAB; KARGAHI, 2020) have proposed a timing error analysis less accurate than GLS simulation by employing learning methods focusing only on functional units, i.e., combinational circuits. This approach prevents these frameworks from evaluating timing errors in hardware accelerators and their persistent internal states in runtime to support sequential circuits. Further, the modeling in (JIAO et al., 2018) is totally agnostic to the temperature and aging degradation effects on timing errors. The modeling in (MOGHADDASI; NASAB; KARGAHI, 2020) considers the transistor's aging degradation isolated from temperature.

The work in (AMROUCH et al., 2019) has analyzed the GLS at the circuit level using an offline approach where the timing errors are propagated as faults to the algorithm and application layers. The offline method employed in (AMROUCH et al., 2019) reduces the scope to embrace open-loop hardware accelerator-algorithm interactions, without covering closed-loops. Similar work is presented in (AFZALI-KUSHA et al., 2020) which evaluates timing errors of an accuracy-configurable approximate multiplier under aging effects and voltage reduction within a framework covering only open-loop applications.

A comprehensive degradation modeling is proposed in (AMROUCH et al., 2019), where the authors explore a joint aging-temperature degradation cell library concept. The modeling in (AMROUCH et al., 2019; JIAO et al., 2018; ZERVAKIS et al., 2018; MOGHADDASI; NASAB; KARGAHI, 2020; HE; GERSTLAUER; ORSHANSKY, 2013) are based on a predictive technology model (PTM) for a 45nm hypothetical bulk process. The work in (WANG; ROBINSON, 2019) employs Synopsys' standard cells for a 32nm hypothetical bulk process. The standard cells in (AFZALI-KUSHA et al., 2020) are built including aging models for a 15nm PTM FinFet process.

Distinction from state-of-the-art: Our work supports the investigation of any degradation-induced timing impacts on the joint accelerator-algorithm in-the-loop within the application. It does so by analyzing the runtime behavior of relevant algorithms. Our work bridges the large gap between the gate-level simulation (GLS) and the application software dynamically for degradation-induced timing errors investigation, embracing complex algorithms with one or more closed-loops. Unlike all works, our holistic framework employs accurate models for the degradation-induced effects based on fully calibrated 14nm FinFET device measurements from Intel technology.

5.4.2 Results Evaluation and Discussions

This section presents the results regarding the complex interactions between the executed algorithm and errors induced by degradation effects underlying hardware of accelerators. We evaluate the framework presented in Section 3 for three existing motion estimation BMAs processed by a SAD accelerator into the HEVC application case study detailed in Section IV.

5.4.2.1 Implementation Details

The implementation considers industrial tools widely employed in standard cell hardware design flow. However, the implemented framework is *tool agnostic* and can straightforwardly be adapted to any RTL synthesis and GLS tool flow. SAD hardware accelerator has parallelism to process eight comparisons per clock cycle. The architectures were described in VHDL with the adder inferred by the synthesis tool ('plus' operator in VHDL). Cadence Genus$^{\text{TM}}$ tool was used to perform the RTL to gate-level netlist synthesis and the STA. The hardware accelerator was mapped in design-time to a 14 nm FinFet standard cells for 0.8V voltage supply and temperature of 25°C, aiming for its maximum frequency of 1.436 GHz (with zero slack value). Cadence Incisive$^{\text{TM}}$ simulator was used to perform the gate-level netlist simulations considering the delays. We evaluate three different degradation cases, considering the temperature increasing to (a) 50°C (25°C above the design); (b) 75°C, to simulate the maximum bound of a mobile device, and with 10 years aging for both (c) 50°C and (d) 75°C to investigate the impact at long-term.

The results evaluation and the discussions for this case study implementation are presented in the following sections. We evaluate the joint accelerator-algorithm runtime behavior encoding 50 frames of the BasketBallPass 416×240 pixels video sequence at a rate of 50 frames per second (fps) for four QPs, using the x265 HEVC encoder. We ran the encoder using the superfast encoder preset with one reference frame to simulate mobile systems' use, keeping the highest coding tree unit (64×64) on the search.

The adequate evaluation to be performed regarding the BMA is to use a metric that associates quality with compression. Therefore, we analyzed four quantization parameters (QPs) 22, 27, 32, and 37, as stated by the Common Test Conditions (CTCs) (BOSSEN et al., 2013) recommendation. The runs with these four QPs generate the four points required to generate BD-BR and BD-PSNR, which are coding-efficiency measurement metrics specific for video compression technology. BD-BR indicates how higher or lower is the bitrate should be to achieve the same PSNR quality. Positive values indicate an increase in the bitrate, which is not desired.

5.4.2.2 Critical Path Delay Analysis

Table 5.5 shows the CPD analysis and the respective error-free maximum clock frequencies for each evaluated case. The CPD of the synthesized SAD netlist was evaluated employing the STA with our degradation-aware standard cell libraries, which contain the cells' delay information. In this work, we evaluate the SAD operation under degradation-induced timing errors at 1.436 GHz (i.e., the maximum error-free frequency for the baseline temperature at 25°C and the fresh circuit). The temperature rising from 25°C to 50°C increases the CPD in +7.47%, and for 75°C +12.06%, respectively. In the presence of 10-Y aging effects, the CPD increases in +15.23% in 50°C and +21.12% for 75°C. Hence, the error-free maximum clock frequency decreases in -6.96% at 50°C temperature, and at 75°C in -10.72%, respectively. For 10-Y aging, the error-free maximum clock frequency reduces in -13.23% at 50°C temperature, and in -17.41% at 75°C. The temperature influences the aging process during circuit life. We considered that the aging processes were accelerated by an average temperature of 50°C during the device's life.

Table 5.5: Critical path delay, error-free clock frequency of the SAD accelerator for each runtime condition.

Runtime Conditions		**Critical Path Delay**	**Error-Free Maximum Clock Frequency**
Temp.	**Aging**	(ps)	(MHz)
25°C	No	696	1.436 GHz
50°C		748 (+7.47%)	1.336 GHz (-6.96%)
75°C		780 (+12.06%)	1.282 GHz (-10.72%)
50°C	10-Y[1]	802 (+15.23%)	1.246 GHz (-13.23%)
75°C		843 (+21.12%)	1.186 GHz (-17.41%)

[1] 10-year aging considering the average temperature of 50°C.

5.4.2.3 SAD Hardware Accelerator Runtime Bit Error Rate

Degradation-induced timing errors are higher towards the MSBs due to the adders' CPDs. However, the exact bit position where the errors are most likely to appear is highly dependent on the input data value range and the hardware accelerator details. Hence, Fig. 5.11 shows the SAD bit error rate (BER) caused by degradation-induced timing errors, considering the average frame error in each output bit position for the three evaluated algorithms. The BER metric used here solely aims to illustrate which parts of the hardware accelerator are more sensitive to device degradation and how the comparison decisions of different algorithms can affect the perceived error. For a precise bit position error estimation, we consider the instantaneous error on the SAD regarding the current inputs, i.e., we compute the expected SAD value regarding the pixel blocks and the previous accumulator value and subtract it from the current SAD hardware accelerator output. For instance, this instantaneous error is the partial SAD value represented in the example of Fig. 5.10(a).

Note that Fig. 5.11 represents the average BER in quantity being accumulated at every single cycle (as in Fig. Fig. 5.10(a)), instead of the total already accumulated value (Fig. 5.10(b)). Using this difference, we identify which bits were flipped, and we update the bit error counter according to the error position.

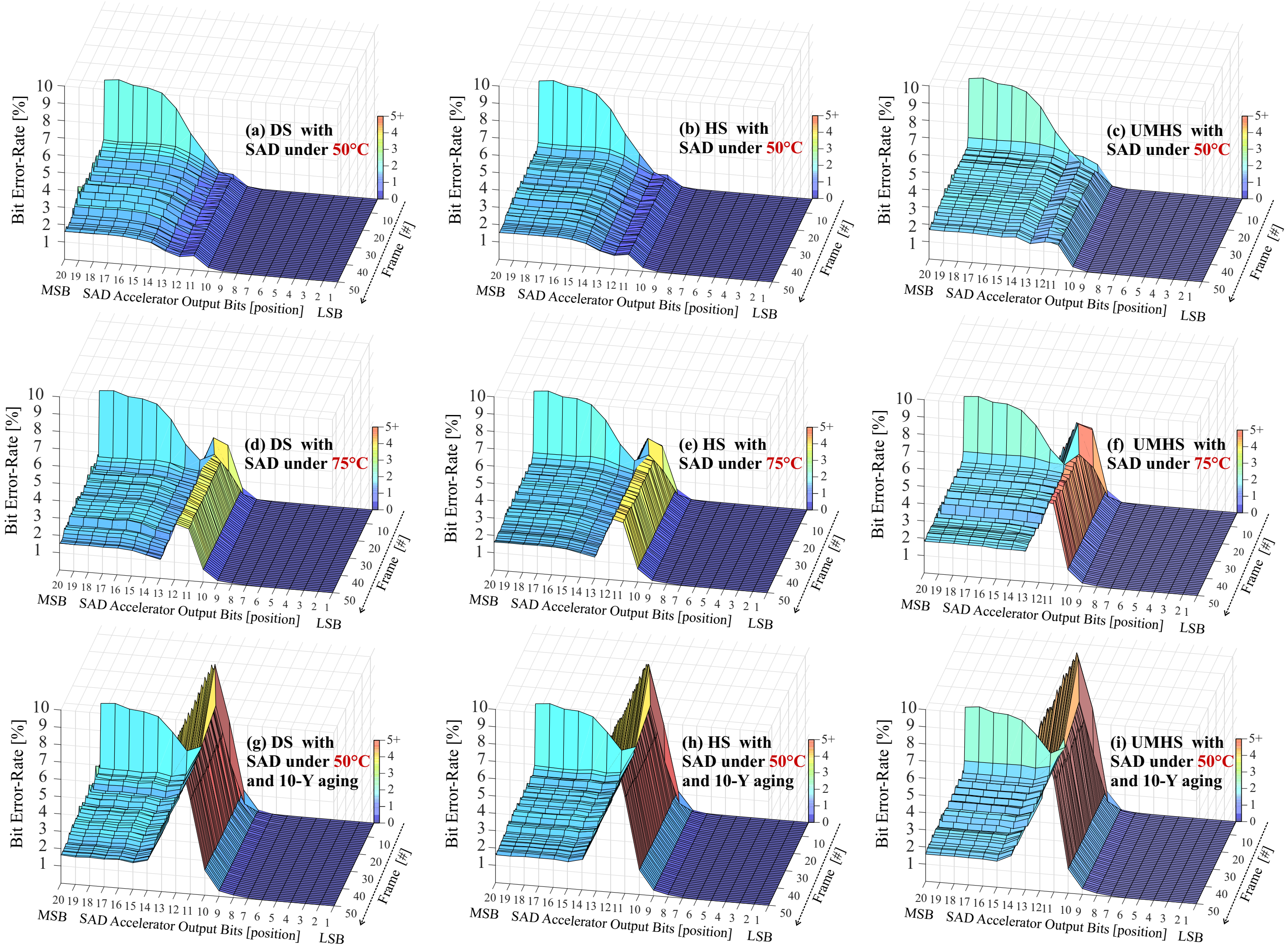
The difference between the compared blocks is higher at the beginning of the motion estimation search; therefore, this causes higher SAD BER for all algorithms in the first frame (Fig. 5.11(a-c)). Further, Fig. 5.11 demonstrates that the three different algorithms result in different SAD BER as a result of the various comparison decisions.

In the DS algorithm with the SAD accelerator temperature rising from 25°C to 50°C (Fig. 5.11(a)), the errors begin to appear on the 9th and 10th bits and smoothly increase to the MSBs. The BER increases in the 75°C case comparing to 50°C. When considering the temperature rising to 50°C and 10-year aging effects (Fig. 5.11(g)), the BER rises sharply in the bit positions from 10th to 11th with a peak of up to 10% of BER in the 11th.

BER behavior of the HS is almost the same as the DS algorithm. Nonetheless, the UMHS algorithm has higher BER in multiple bit positions (11th and 12th) as it performs a broader search to move away from the global minima, as shown in Fig. 5.11(i). The probability of a more considerable SAD difference is higher when the algorithm tries to compare block pixels, which are far from the current reference position, which may increase the error and lead the UMHS to look even further, creating a snowball effect on the accumulated error, which increase the operand. Note that the BER in the UMHS algorithm case is the worst in all cases at the same accelerator's operational conditions.

Observing the SAD architecture in Fig. 5.3, most of the time, the multiplexer selector is in logic 1 (accumulating). Note that, in the SAD hardware accelerator, the last adder's bit-width is wider (20-bit) because it is an accumulator. This last adder receives a faster input with 20 bits (i.e., wire from the register) and a slower input with 11 bits (from the adder three). Therefore, in Fig. 5.11, the eight MSBs from the output are less affected (about 2% in all cases), occurring only due to the errors in the carry propagation of the last adder. The seven least significant bits (LSBs), in Fig. 5.11, are not affected in all cases due to their shorter timing paths where there are no timing errors. Then, the higher timing errors are concentrated between the output bits 8th to 13th.

Figure 5.11: Bit error rate of SAD per bit for each frame under degradation during 50 frames of the BasketBallPass 416×240 video sequence with QP=32 for algorithms: DS (a,d,g), HS (b,e,h), UMHS (c,f,i); with three cases of SAD delay degradation: 50°C of temperature (a-c), 75°C (d-f), 50°C combined with 10-year aging (g-i).



Source: The Author.

5.4.2.4 Algorithm Runtime SAD Executions Impact

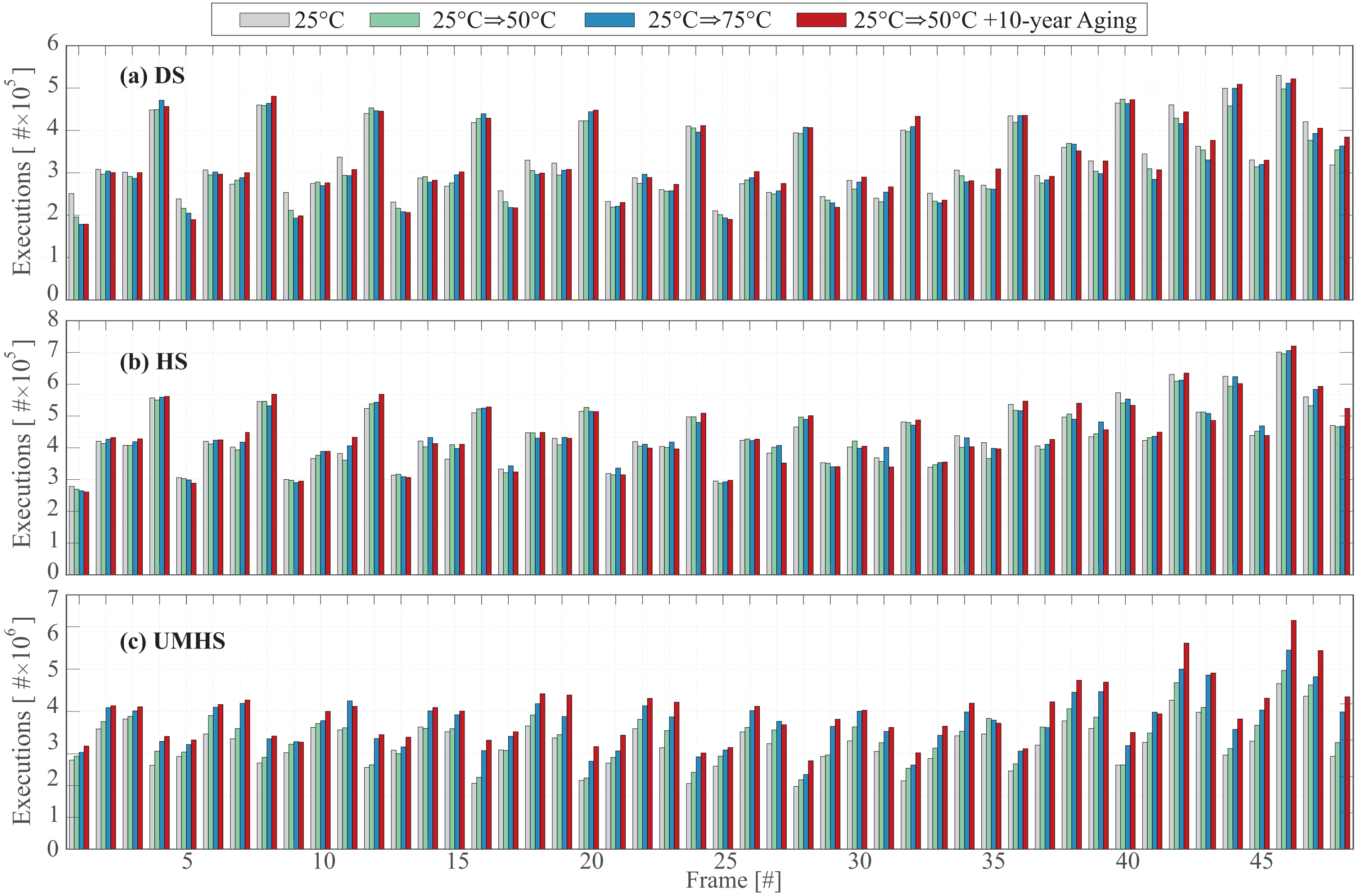
Fig. 5.12 shows the number of SAD executions per frame to encode 50 frames of the BasketballPass sequence at 50 frames per second. We present the default case, which is the gray bar representing the simulation at 25°C, along with three different situations, as mentioned in Section 5.1. The results are presented for the three BMAs mentioned earlier (DS, HS, and UMHS) responsible for the HEVC IME stage, considering the impacts of the device degradation in the SAD hardware accelerator.

The DS algorithm fetches four neighbor candidate blocks in each interaction, and when neither candidate is better than the current center, it terminates the search. The degradation-induced noise tends to make the best candidate, which has suffered the least degradation that causes the SAD value to decrease. Therefore, the DS algorithm tends to stop earlier in the presence of high - magnitude noise because, when it finds a block with a smaller difference value, it tends to look around without finding a more similar block (i.e., with a SAD value lower than the current center). We can observe from Fig. 5.12(a) that the degradation effects into SAD cause an early termination that tends to slightly reduce the number of the executions, in general, for the presented temperature and aging scenarios. Specific cases that lead to an increase in the number of executions in the DS BMA (in some frames) are due to errors that may increase the candidate's SAD value that would be a good match in the scenario without degradation and can make the BMA perform more iterations.

Fig. 5.12(b) shows the degradation scenarios for the HS BMA. Even though the HS tends to present a similar number of executions when increasing the temperature from 25°C to 50°C and 75°C, a slight increase is observed when increasing it to 50°C and considering 10-Y aging. The increases and decreases can be explained by the HS flow similarly as in DS: the HS main stage (the Hexagon step itself) stops earlier when a SAD calculation under degradation deviates from its precise value and generates a smaller SAD. In these cases, the Hexagon step does not find any better candidates and executes the square refinement step, which has a fixed number of candidates, so degradations occurring in that step will not affect the number of SAD calculations for that specific execution.

The UMHS BMA presents a more straightforward result in terms of SAD executions, given that, in a large majority of the cases, the number of executions tends to increase with a temperature rise and with the inclusion of aging, as shown in Fig. 5.12(c). It mainly occurs because UMHS is sensitive to the chosen candidates in several stages of the search. First of all, the algorithm contains a few stop conditions at the beginning of its execution when the best current SAD is smaller than predefined thresholds. Under degradation, when the architecture presented a SAD value that is much higher than the correct one for a specific candidate, an execution that would initially stop due to threshold conditions may continue its execution, calculating SAD for several other candidates. Another reason for UMHS to execute SAD for more candidates is that certain conditions may lead it to execute the whole Hexagon BMA after the normal execution of UMHS, as mentioned in section V. Additionally, the increase in the SAD executions in UMHS under degradation may be specifically harmful to the application because it may aggravate the system's timing requirements, which may be incapable of achieving real-time throughput.

Figure 5.12: Number of SAD executions per frame under degradation for 50 frames of the BasketBallPass 416×240 video sequence and the three algorithms: (a) Diamond search (DS), (b) Hexagon search (HS) and (C) Uneven multi-hexagon search (UMHS).



Source: The Author.

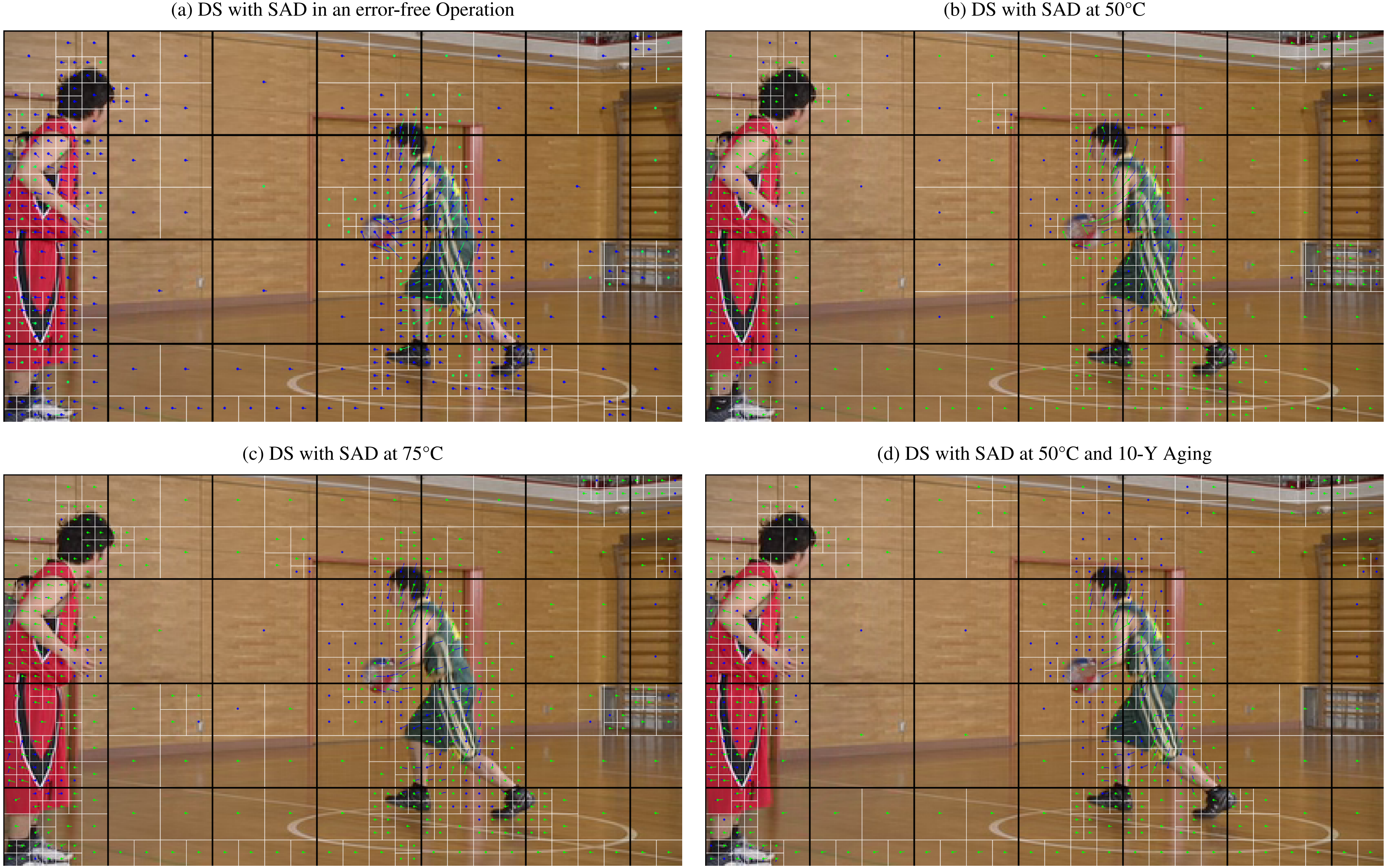
5.4.2.5 Coding Impacts Analysis

The outputs of ME algorithms consist of MVs – pointing to the best block match – and the PU modes partitioning. Figs. 5.13-5.15 show the results for subjective quality, MVs, and PU partitioning analysis of a high-motion scene of BasketballPass video sequence (frame number 46), for the BMAs DS and UMHS, respectively. Note that the frame analysis was performed for the same video and encoder configurations in all analyses. The visual results were restricted to DS and UMHS algorithms (the best and worst in the reliability, respectively) at QP 32 to reduce the number of pictures.

Figs. 5.13 and 5.15 show the final video image quality with the SAD under different degradation phenomena and its interaction with the DS and UMHS algorithms, respectively. We can subjectively view that the visual quality was not substantially affected by any degradation-induced timing-errors evaluated. The cases evaluated – from (a) to (d), in Figs. 5.13-5.15 – did not result in an image artifacts. It demonstrates the prediction algorithms' resilience in tolerating degradation-induced timing relaxation errors on the SAD hardware accelerator design. However, compression efficiency can be compromised by spending a higher bit rate to maintain quality.

The black-colored lines in Figs. 5.13-5.15 delimit the CTUs, which were set to a fixed size of 64×64, and the blank-colored lines represent the selected PU partitions for each CTU. The partition decisions remained almost the same when considering the HS and DS BMA up to 50°C of temperature. When increasing the temperature to 75°C, we observe a few new sub-partitions close to both basketball players' feet.

Figure 5.13: Motion vector analysis to evaluate the degradation impact in a high-movement scene during the HEVC encoding: DS, (a) The golden model, (b) Operating at 50°C, (c) Operating at 50°C and (d) Operating at 50°C with 10-year aging.



(a) DS with SAD in an error-free Operation

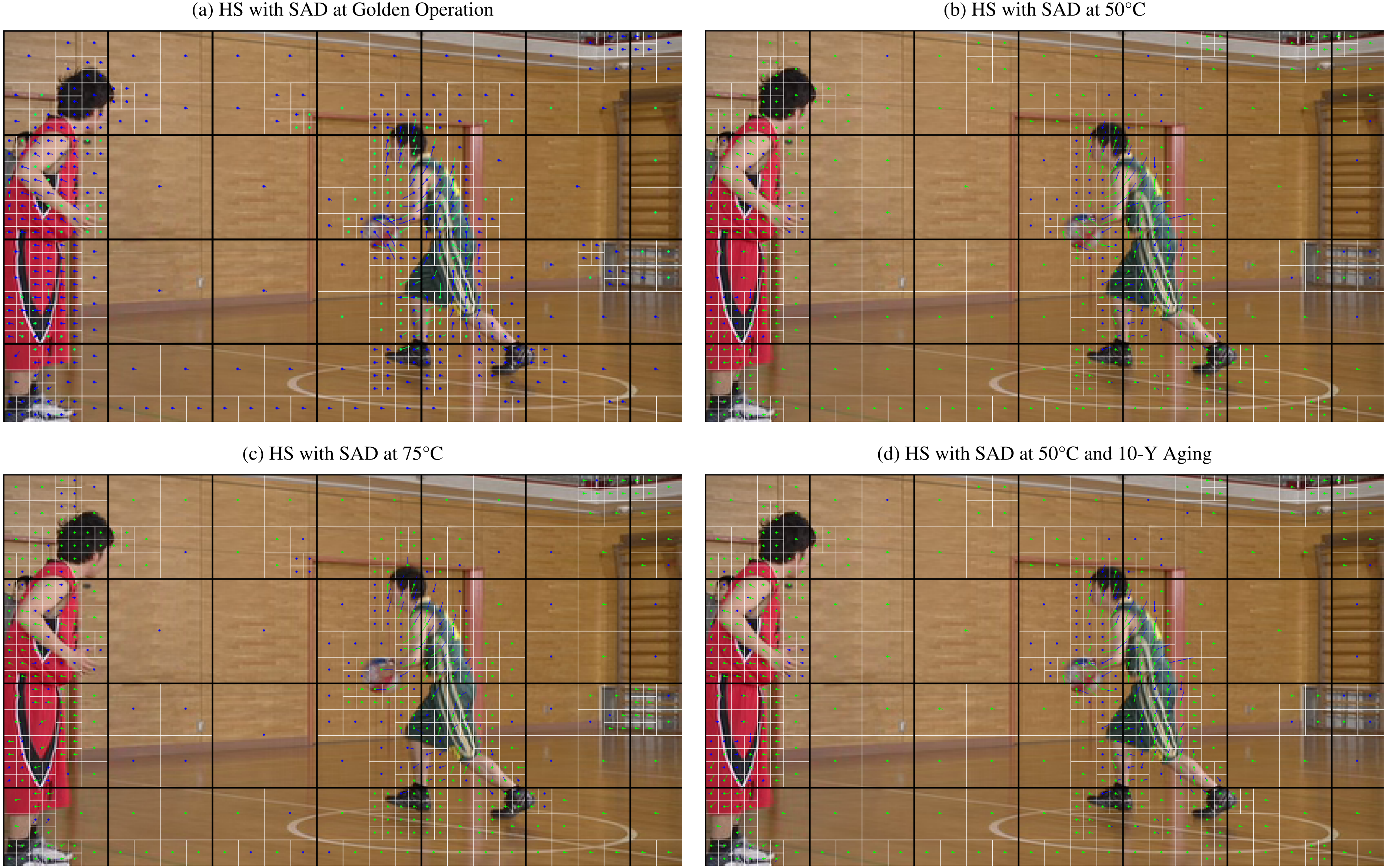
(b) DS with SAD at 50°C

(c) DS with SAD at 75°C

(d) DS with SAD at 50°C and 10-Y Aging

Source: The Author.

Figure 5.14: Motion vector analysis to evaluate the degradation impact in a high-movement scene during the HEVC encoding: HS, (a) The golden model, (b) Operating at 50°C, (c) Operating at 50°C and (d) Operating at 50°C with 10-year aging.



(a) HS with SAD at Golden Operation

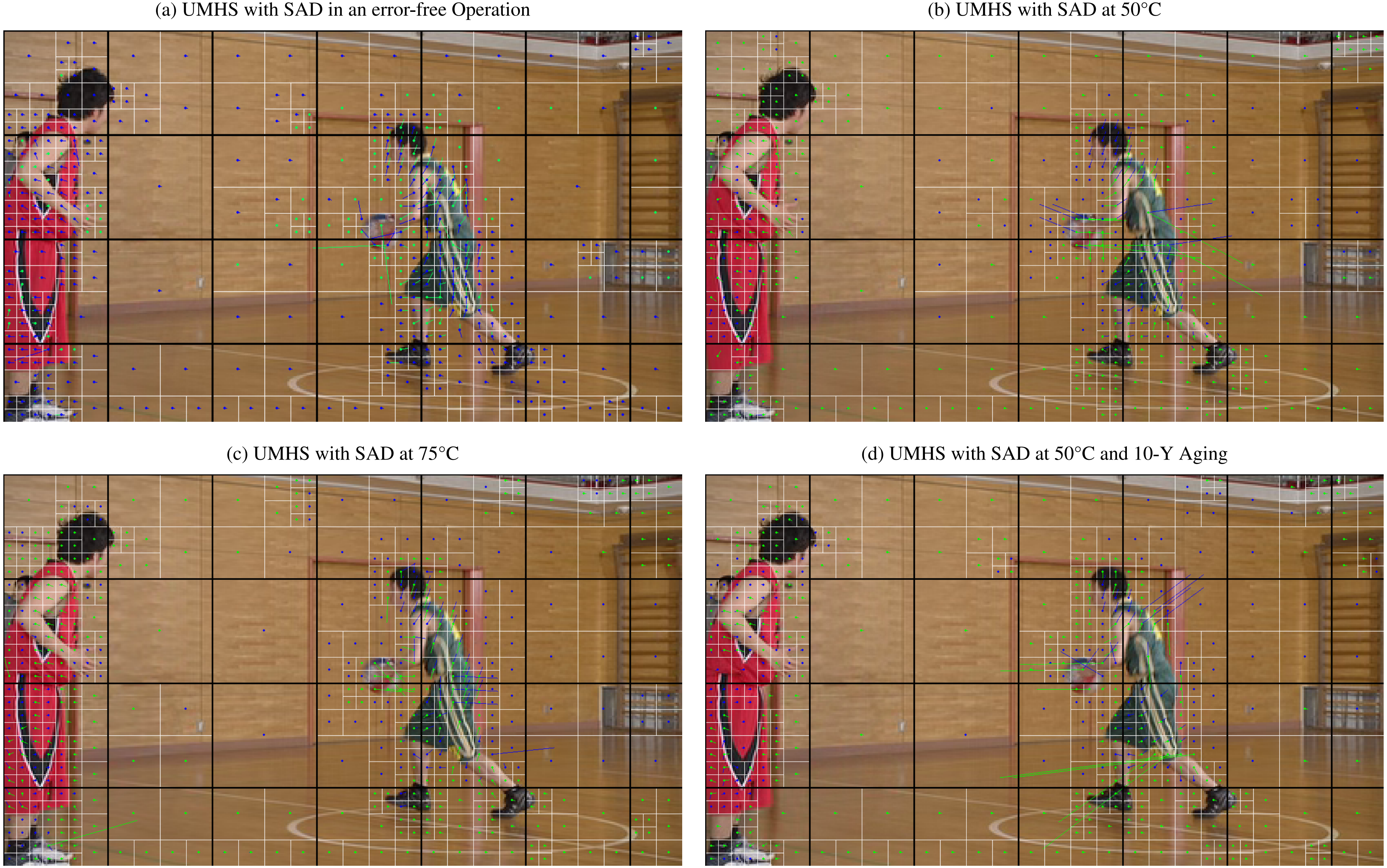
(b) HS with SAD at 50°C

(c) HS with SAD at 75°C

(d) HS with SAD at 50°C and 10-Y Aging

Source: The Author.

Figure 5.15: Motion vector analysis to evaluate the degradation impact in a high-movement scene during the HEVC encoding: UMHS, (a) The golden model, (b) Operating at 50°C, (c) Operating at 50°C and (d) Operating at 50°C with 10-year aging.



(a) UMHS with SAD in an error-free Operation

(b) UMHS with SAD at 50°C

(c) UMHS with SAD at 75°C

(d) UMHS with SAD at 50°C and 10-Y Aging

Source: The Author.

The UMHS BMA presented more observable effects on the partitioning, deciding to split some CTUs in the middle of the image, as can be seen from the difference between Fig. 5.15(b) and 5.15(c). Moreover, the changes are even more severe, in terms of partitioning, when considering the 50°C temperature along with 10-Y aging, as shown from the differences between the scenarios depicted in Figs. 5.15(a-c) and 5.15(d).

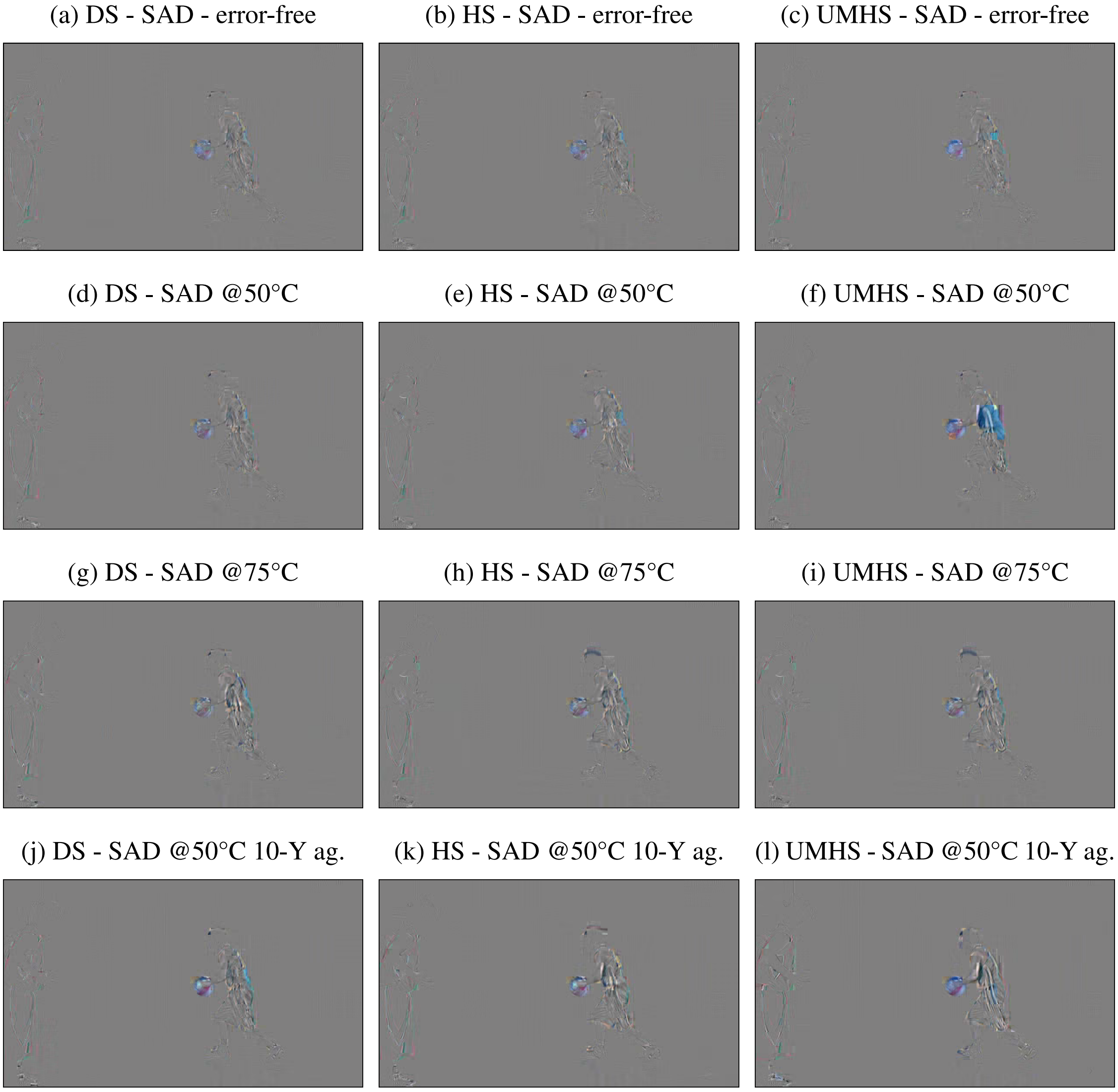
5.4.2.6 Motion Vectors

Figs. 5.13-5.15 also present the s of each PU, where blue-colored lines represent L0 MVs, and the green-colored are for L1 MVs. The L0 and L1 types are directly related to which reference frame the best MV comes from. The points represent the beginning of the vectors. The MV analysis results show that up to a temperature rise of 50°C on the SAD accelerator (presented in Figs. 5.13(b), DS BMA seems to tolerate the degradation-induced timing errors and tends to maintain similar vector decisions. When the temperature increases up to 75°C, DS BMA (Figs. 5.13(c)) still seems to tolerate the errors, presenting slight differences in the type between the MVs obtained by the search. However, the UMHS algorithm begins with severe changes at 50°C, as several of the lines have changed their colors from Fig. 5.15(a) to Fig. 5.15(b). At the temperature of 75°C, Fig. 5.15(c) shows that all the BMA considered having changed considerably regarding MV type when compared to the golden model, less than 50°C along with 10-Y aging. The worst case is the UMHS at 50°C along with 10-Y aging of the Fig. 5.15(d) presenting the higher MVs pointing to the wrong places.

5.4.2.7 Prediction Residue Impact

The result of the prediction encoding tool is the predicted block, which will be used to generate the residue. Therefore, we include the residual frame to perform a subjective analysis – such as the one from Figs. 5.13-5.15 – of the prediction impacts when operating on temperature rising and aging degradation. Fig. 5.16 shows the prediction residue impacts for the same three different algorithms (DS, HS, and UMHS) on each temperature and aging degradation case, including the normal operation scenario for comparison purposes. The same frame and video from the previous analysis were selected for the prediction analysis of Fig. 5.16 and higher motion activity and higher stress to the motion estimation algorithms. The first three frames (Fig. 5.16(a-c)) demonstrate the result for the SAD with a normal operation for the three algorithms.

Figure 5.16: Prediction residue results due to the degradation-induced timing-errors on SAD impact on the three algorithms DS, HS, UMHS.



(a) DS - SAD - error-free (b) HS - SAD - error-free (c) UMHS - SAD - error-free

(d) DS - SAD @50°C (e) HS - SAD @50°C (f) UMHS - SAD @50°C

(g) DS - SAD @75°C (h) HS - SAD @75°C (i) UMHS - SAD @75°C

(j) DS - SAD @50°C 10-Y ag. (k) HS - SAD @50°C 10-Y ag. (l) UMHS - SAD @50°C 10-Y ag.

Source: The Author.

Grey-colored pixels mean that the residue is zero. Therefore, the prediction stage estimates all the necessary information to reconstruct these pixels. The background of the basketball scene (see Fig. 5.15) performed well in the prediction stage, resulting in a major part of grey-colored pixels in the residue from Fig. 5.16. However, we can notice that a player is moving with a basketball in the scene. This movement makes the prediction difficult, as there is a slighter possibility of finding a similar block matching. Therefore, the residue increases, mainly in the players' edges and the basketball itself. We can observe that temperature rising from 25°C to 50°C (Fig. 5.16(d-f)) does not present noticeable impacts on the image residue for the DS and HS algorithms. However, the prediction stage using the UMHS BMA demonstrates a high impact in the uniform number on the player's back with the ball. The 75ºC scenario (Fig. 5.16(g-i)) presents increased residue in the right player's arms and legs comparing with 50°C. The case combining 50°C temperature and 10-year aging (Fig. 5.16(j-l)) reveals that the algorithms increase the residue on the player region in all cases. However, the UMHS algorithm maintains a high prediction error due to the worst MVs choices observed in the previous analysis (see Fig. 5.15).

In the prediction residue, we can observe from the scene background that the decisions in this region were maintained. The background preservation happens because this region is usually decided in a few interactions and uses the lower part of the SAD that was not impacted by timing errors. However, the UMHS algorithm demonstrates a non-reliable behavior, resulting in a higher prediction residue impact in all cases. We observe that the presence of degradation effects on the SAD accelerator tends to dazzle the decisions of the IME BMAs.

5.4.2.8 Compression Efficiency Impact

Further, we evaluate the impact of video compression efficiency using two measurement methods specific to video compression technology: BD-BR and BD-PSNR. Table 5.6 summarizes the compression efficiency and the processing overhead (PO) on average results between the video frames (see Fig. 5.12).

The most reliable BMA for the IME in all cases is the DS. In our first evaluated case study with SAD operating at 50° (25° above the design), the DS shows a modest increase of +1.98% on BD-BR and a negligible drop in BD-PSNR compression efficiency metrics. In the 50° case, the DS algorithm is also faster than standard with a 3.36% fewer executions, on average. In the temperature rising to 75°, the DS algorithm increases to 8.85% the BD-BR and drops about 0.4162dB in BD-PSNR without PO 2.765% less executions on average. After 10 years of operation considering an average temperature of 50° during the aging process, if running at 50° in runtime, the DS increases to about 12.87% the BD-BR, with 0.5898dB of BD-PSNR compression efficiency drop. For the same 10-year aging scenario, in the worst runtime temperature case evaluated when rising to 75°, the DS increases the BD-BR by about 19.23%, and the BD-PSNR drops -0.864dB remaining with a negligible PO.

UMHS is the worst algorithm in all cases, showing that it is unreliable in the presence of timing errors. When operating at a 50° temperature, UMHS starts with a 5.7% of BD-BR loss and 0.27 17dB drop in the BD-PSNR with a PO of about 8.06%, which is higher than the boost of +6.96% maximum clock frequency provided by the

Table 5.6: Impacts summary of temperature rising and 10-year aging effects on SAD accelerator for each algorithm.

Runtime Conditions		IME Algo.	Processing Overhead	Coding-Efficiency	
				$^{\alpha}$BD-BR	$^{\beta}$BD-PSNR
Temp.	**Aging**		(%)	(%)	(dB)
50°C	No	**DS**	-3.364	+1.983	-0.0967
		HS	-0.804	+2.602	-0.1257
		UMHS	+8.059	+5.706	-0.2717
75°C	No	**DS**	-2.765	+8.849	-0.4162
		HS	+1.045	+9.536	-0.4462
		UMHS	+21.658	+19.280	-0.8653
50°C	10 Years[1]	**DS**	-0.488	+12.877	-0.5898
		HS	+1.487	+14.520	-0.6661
		UMHS	+29.875	+26.299	-1.1568
75°C	10 Years[1]	**DS**	+0.015	+19.225	-0.8642
		HS	+5.300	+21.856	-0.9707
		UMHS	+46.956	+37.361	-1.5719

*The anchor is the same algorithm with error-free operation.

$^{\alpha}$Positive BD-BR means the increase on the stream bitrate for the same quality.

$^{\beta}$Negative BD-PSNR means the decrease on quality for the same bitrate.

[1] 10-year aging considering the average temperature of 50°C.

timing guardbands removal at 50°. UMHS algorithm also demonstrates a high loss for 75° with about 19.28% of BD-BR increase and a -0.865 3dB drop in BD-PSNR with 21.6 6% PO, (which doubles the overhead of the 10.72% maximum clock frequency boost provided). Note that the UMHS algorithm demonstrated to be unreliable and did not tolerate temperature-induced timing errors, even before the inclusion of aging effects. The 10-year aging at 50° for UMHS was worst than 75° with the fresh circuit, increasing the BD-BR in 26.30% and dropping the BD-PSNR in 1.1568dB with a PO of 29.88%.

HS algorithm is better than UMHS and worst than DS, increasing 2.6% in BD-BR in 50°, with 0.012 57dB in BD-PSNR that is considered a low-impact, without impacts the number of executions. When the SAD is operating at 75°, the HS algorithm shows a 9.536% loss in BD-BR and 0.446dB drop in BD-PSNR with negligible PO. After the 10-year aging, operating at 50°, the HS increase the degradation to 14.52% of BD-BR increase with 0.6661dB drop in BD-PSNR with also a negligible PO. In the worst-case evaluated scenario, with 10-year aging operating at 75°, the HS resulted in about 21.86% of BD-BR increase with -0.9707dB drop in BD-PSNR with a PO of about 5.30%.

The compression efficiency is highly impacted when employing the UMHS, causing up to 37.36% of higher transmission data in the video streaming. Also, the desired improvements in the maximum clock frequency by removing the guardbands are lost when choosing the UMHS because its processing overhead is always higher than the frequency boost (see in Tables 5.5 and 5.6). Note that DS is the most reliable between the algorithms in terms of ultimate compression efficiency and demonstrates a negligible PO even in the worst-case scenario.

The least impacted algorithms by the degradation-induced timing errors are HS and DS, based on groping the surroundings with smaller steps. We can interpret this behavior as the metaphor of an aged, almost blind person with a cane that prefers to perform smaller steps with less risk of going in the wrong direction than an imperfect measured long step. The metaphor, as mentioned earlier, helps to understand the UMHS algorithm behavior under degradation that demonstrates more risk of going in the wrong direction with a long stride and getting lost (see Fig. 2.4), increasing both the interactions and the loss in compression efficiency.

5.4.3 Comparisons to the State of the Art

The modeling in (JIAO et al., 2018) was totally agnostic to the temperature and aging degradation effects on timing errors. Therefore, the work in (JIAO et al., 2018) did not consider that the degradation effects differently affect each gate's delay, as demonstrated in (AMROUCH et al., 2019) and herein also fully considered. The models in (HE; GERSTLAUER; ORSHANSKY, 2013; ZERVAKIS et al., 2018) did not evaluated temperature and aging on timing error effects. The modeling in (MOGHADDASI; NASAB; KARGAHI, 2020) considers the aging effects. However, it did not consider the temperature degradation effects. In (AMROUCH et al., 2019) and (MOGHADDASI; NASAB; KARGAHI, 2020), the modeling were based on a predictive technology model (PTM) for a 45nm hypothetical bulk process. The work in (WANG; ROBINSON, 2019) employed the Synopsys' standard cells for a 32nm hypothetical bulk process. The standard cells in (AFZALI-KUSHA et al., 2020) were built, including aging models for a 15nm PTM hypothetical FinFet process. This case study employed accurate device models for the degradation-induced effects based on a fully calibrated 14nm FinFET device measurements from Intel technology.

5.4.4 Conclusions of the Section

This section employed the framework proposal for crossing temperature-induced timing errors from the underlying hardware accelerator to the algorithms and application layers. It demonstrates the complex and dynamic interaction between the algorithms and the underlying timing errors at runtime. Our work employs degradation-aware cell libraries for aging and temperature effects to achieve this goal while crossing the layers simulating the gate-level dynamically into the application. The framework proposal was herein generically demonstrated as being compatible with any application and any hardware accelerator. We evaluate the framework by investigating the effects using an existing real-world HEVC video compression application. Our results introduce the ultimate impact of degradation-induced timing errors underlying hardware accelerator when interacting with the algorithms. Our results quantify the degradation effects impact on processing overhead, compression efficiency, and video quality. Between the algorithms evaluated, the DS algorithm is the most reliable and is capable of enduring 50°C with less than 2% of BD-BR increase and in the worst-case, 19.22% of BD-BR increase under 75°C with 10-Y aging. On the other hand, we also reveal that the UMHS is an unreliable algorithm with a 5.71% BD-BR increase under 50°C and up to 37.36% in the worst-case investigated at 75°C with 10-Y aging. Our case study demonstrated the virtue of our holistic framework investigating the impacts on the algorithms' runtime. The results show that the boost of about 6.96% (at 50°) to 17.41% (at 75° with 10-Y aging) in the maximum clock frequency achieved by removing the timing guardbands is totally lost by the processing overhead from 8.06% to 46.96% caused by the UMHS algorithm due to its interactions with the SAD accelerator under timing errors. Unlike the UMHS algorithm, the DS algorithm's reliable behavior was revealed by our framework showing a negligible processing overhead for all cases.

5.5 Case Study III: Voltage Over-scaling

Voltage Over-Scaling (VOS) optimizes energy while causing timing errors due to an unsustainable clock frequency. Many algorithms, such as in multimedia and machine learning applications, are capable of tolerating such errors. VOS has never been investigated in hardware accelerators running closed-loop algorithms. As the errors impact most decisions and actions in the subsequent steps, closed-loops dynamically change the execution flow. Timing errors should be evaluated by an accurate gate-level simulation. However, a significant gap still remains: how these timing errors propagate from the underlying hardware all the way up to the entire algorithm run, where they just may degrade the performance and quality of service of the application at stake? This thesis tackles this issue showing a framework for VOS investigation, embracing any kind of application. Our framework simulates the VOS-induced timing errors at the gate level, dynamically linking the hardware result with the algorithm and vice versa during the evolution of the runtime of the application. The state-of-the-art VOS literature for video encoding application fails to assess the ultimate impacts of VOS-induced timing errors, as current works open the encoding loops. Unlike those, our work investigates the ultimate impact of a hardware accelerator dynamically carrying through to the video encoder all VOS-induced timing errors and preserving the full compliance to the standard.

We purposefully chose the SAD hardware accelerator and the motion estimation algorithm as a case study to demonstrate how VOS-induced timing errors impact closed-loop applications. We employ a parallel sum of absolute differences (SAD) hardware accelerator as a case study. We assess the performance of the overall encoder under varying timing guardbands. Thus, we can evaluate the effectiveness of the implemented method in an industrial-strength accelerator-algorithm co-design. We evaluate the VOS in the video encoder employing the cross-layer method presented in the previous section. For our analysis, we consider the SAD-8 accelerator since the most time-consuming and compute-intensive part of the HEVC is the motion estimation (SILVEIRA et al., 2017) that is mainly based on the sum of absolute differences computation. Then, in our case study, SAD operates in a voltage island (i.e., in other power domains), as in industrial chip implementation (MYERS et al., 2016). Next, it is demonstrated that, under VOS, the ultimate impact in compression efficiency is related to the video's motion intensity. Additionally, the advantages of timing guardband controlled reduction are clearly quantified in our results by virtue of the framework. Reducing at a maximum of 9.5% the clock frequency, energy savings (up to 16.5% in energy/operation) are achieved in SAD for video compression.

5.5.1 Related Work

Most works that exploit tradeoffs between errors and hardware efficiency focused on investigating isolated layers without connecting the hardware accelerators with the algorithm and application layers. A comprehensive survey about this issue is presented in (STANLEY-MARBELL et al., 2020), showing that challenge number one is the quest for holistic cross-layer approaches.

Video accelerators are intrinsically resilient to errors, especially in video compression, where the search for redundancy to compress the video is inherently based on many sub-optimal decisions (PAIM et al., 2020). As a result, video accelerators are perfect use cases for evaluation due to their ubiquitous exploitation, power-constrained execution, and inherent error resilience.

Several works have proposed to lower the voltage of hardware accelerators aiming at its benefits in energy efficiency, as in (ENOMOTO; KOBAYASHI, 2013; SOARES et al., 2019). Nevertheless, few works have investigated the timing error impacts when over-scaling the voltage (VARATKAR; SHANBHAG, 2008; HE; GERSTLAUER; ORSHANSKY, 2013; AMROUCH et al., 2019; AFZALI-KUSHA; KAMAL; PEDRAM, 2020; AFZALI-KUSHA et al., 2020). The works in (LIU; ZHANG; PARHI, 2010; LIU; PARHI, 2011) present the VOS energy-quality tradeoff into standalone arithmetic operators.

VOS in open-loop standalone implementations were explored in finite impulse response (FIR) filters in (LIU; ZHANG; PARHI, 2010; CHEN; HU, 2013; SEDIGHI; ANTHAPADMANABHAN; SUVAKOVIC, 2014), coordinate rotation digital computer processor (CORDIC) (LIU; ZHANG; PARHI, 2010), in a low-density parity-check (LDPC) decoder (SEDIGHI; ANTHAPADMANABHAN; SUVAKOVIC, 2014), in a wavelet-based electrocardiogram (ECG) processor (HAN et al., 2016), in neural networks (ZHANG et al., 2018a; AFZALI-KUSHA; KAMAL; PEDRAM, 2020; AFZALI-KUSHA et al., 2020), and in ray tracing in (AMROUCH et al., 2019). A timing errors modeling for LDPC decoding and investigations about voltage supply faults in the reliability context are shown in (BRKIC; IVANIš; VASIć, 2017; BRKIC; IVANIS; VASIć, 2017).

A state-of-the-art framework for VOS in (ZERVAKIS et al., 2018) demonstrated methods to enable gate-level simulations by modeling timing errors, which achieves high accuracy for voltage over-scaling analysis comparing to the spice. The framework in (ZERVAKIS et al., 2018) was adopted in (ZERVAKIS et al., 2019) also demonstrating the combination of VOS with other approximate computing techniques. Unlike our work, which evaluates the ultimate impact in the application, both works (ZERVAKIS et al., 2018; ZERVAKIS et al., 2019) are limited to evaluate errors in the output of standalone hardware accelerators. Frameworks for VOS based in gate-level simulation modeling, as in (ZERVAKIS et al., 2018) and the work herein presented, can additionally employ state-of-the-art hardware-based emulation platforms (ASIC/FPGA) to accelerate the simulation, as in (WEIßBRICH et al., 2019) or commercial versions from Cadence Protium platform and Cadence Palladium Z1.

The work in (VARATKAR; SHANBHAG, 2008) shown an investigation of a standalone motion estimation accelerator under VOS. The main contribution in (VARATKAR; SHANBHAG, 2008) is a noise tolerance technique demonstrating higher energy savings applying VOS for the same error level in the prediction. Despite showing improvements, the simpler open-loop methodology employed to analyzing a standalone motion estimation block impossibilities to access the ultimate impact of the video coding performance, such as quality, bitrate, or coding efficiency.

A motion estimation implementation in open-loop imposes simplifications resulting in a behavior non-compliant with the video coding standard. By offline feedbacking the errors to the encoder (i.e., in open-loop), one causes coding drift errors with mismatches between the coding and the decoding steps – an unacceptable application error. Besides, the open-loop implementation cannot also capture the error propagation between the reference frames. Therefore, an open-loop method is impracticable for video encoding applications to holistically investigate hardware accelerators under VOS for any compliant encoder with any standard.

The limitations of an open-loop implementation are as follows:

(1) Causes decoding mismatch drift error. The result is unpredictable (i.e., decoded video is different from the encoded). It occurs due to an investigation without the decoding loop of the encoding process, agnostic to other algorithms like discrete cosine transform (DCT), quantization (Q), inverse Q (IQ), inverse DCT (IDCT), and context-adaptive binary arithmetic coding (CABAC).

(2) Uses original frames as a reference to predict the motion vectors using data from the original video, i.e., neglecting the error propagation between the frames due to the impacts in quality in the compressed reference frame. In short, after the encoding, it is impractical to have original frames to reconstruct the video.

(3) To infer the quality, it performs an impracticable reconstruction using the original frames – which is impossible to do in a decoder.

(4) Limited to evaluate only the impact in the predictions residue, performing this impact in an unrealistically way (due to limitation mentioned in the previous items 1-3)).

(5) Cannot evaluate the compressed video size and its final quality and, therefore, no firm conclusion arises about the impact of VOS.

Table 5.7: Summary of related work results.

Related Work	A	B	C	D	E	F	G
(OLIVEIRA et al., 2017)	✓	–	–	–	–	–	N/A
(TU et al., 2019)	✓	–	–	–	–	–	N/A
(MOHANTY, 2020)	✓	–	–	–	–	–	N/A
(MASERA; MARTINA; MASERA, 2017)	✓	–	–	–	–	–	✓
(KAMMOUN et al., 2020)	✓	–	–	–	–	–	✓
(ENOMOTO; KOBAYASHI, 2013)	✓	✓	–	–	–	–	N/A
(SOARES et al., 2019)	✓	✓	–	–	–	–	N/A
(LIU; ZHANG; PARHI, 2010)	✓	✓	✓	–	–	–	N/A
(LIU; PARHI, 2011)	✓	✓	✓	–	–	–	N/A
(CHEN; HU, 2013)	✓	✓	✓	–	–	–	N/A
(ZERVAKIS et al., 2018)	✓	✓	✓	–	–	–	N/A
(ZERVAKIS et al., 2019)	✓	✓	✓	–	–	–	N/A
(JEON et al., 2012)	✓	✓	✓	–	–	–	N/A
(CHEN et al., 2014)	✓	✓	✓	✓	–	–	N/A
(HAN et al., 2016)	✓	✓	✓	✓	–	–	N/A
(ZHANG et al., 2018a)	✓	✓	✓	✓	–	–	N/A
(AFZALI-KUSHA; KAMAL; PEDRAM, 2020)	✓	✓	✓	✓	–	–	N/A
(AFZALI-KUSHA et al., 2020)	✓	✓	✓	✓	–	–	N/A
(AMROUCH et al., 2019)	✓	✓	✓	✓	✓	–	N/A
(PAIM et al., 2020)	✓	–	–	–	–	–	N/A
(VARATKAR; SHANBHAG, 2006)	✓	✓	✓	–	–	–	–
(VARATKAR; SHANBHAG, 2008)	✓	✓	✓	–	–	–	–
(HE; GERSTLAUER; ORSHANSKY, 2013)	✓	✓	✓	–	✓	–	–
This work	✓	✓	✓	✓	✓	✓	✓

(A) Trading-off hardware errors to energy efficiency.

(B) Explores energy savings of voltage scaling.

(C) Explores timing errors impacts induced by voltage over-scaling.

(D) Presents the system-level impact of VOS.

(E) Explores timing guardbands for VOS.

(F) Bridging the gap between VOS and joint accelerator-algorithm closed-loop.

(G) Fully compliant with a video coding standard.

The work in (HE; GERSTLAUER; ORSHANSKY, 2013) presents stand-alone DCT/IDCT hardware accelerators under VOS showing results only for JPEG image compression application at the circuit layer (i.e., in open-loop). The work in (HE; GERSTLAUER; ORSHANSKY, 2013) cannot present the ultimate impact analysis in terms of compression efficiency due to the limitation of performing the gate-level simulation (GLS) in an open-loop approach. A direction, as future work highlighted in (HE; GERSTLAUER; ORSHANSKY, 2013) is to evaluate the impact of the timing errors underlying hardware accelerators in the full context of the video compression (i.e., within the encoder's closed-loop).

VOS must be evaluated by a time-accurate gate-level simulation to unveil how the timing errors in the underlying hardware ultimately impact the application level. The cited related works are limited to investigate stand-alone hardware accelerators or directly open-loop applications such as filtering. Applications with closed-loops between the joint accelerator-algorithm (i.e., a connection between hardware-software) have never been investigated for the VOS impact on quality and clock guardbanding. Closed-loops require feedback from a time-accurate gate-level simulation during the application runtime. Recently, (PAIM et al., 2020) has shown a cross-layer co-simulation method to bridge gate-level simulations to evaluate logic errors crossing all the way up to the application quality and energy efficiency. Therefore, supporting VOS-induced timing error feedback in closed-loop applications remains challenging. To solve this challenge, we herein extended the cross-layer framework for treating VOS-induced timing violations as demonstrated in (ZERVAKIS et al., 2018) as well as including the voltage over-scaling standard cell libraries generation as in (AMROUCH et al., 2019).

Distinction from the VOS works of the state of the art: The framework herein demonstrated enables the investigation of VOS-induced timing errors in the complex iterations of the joint accelerator-algorithm running dynamically at application runtime. The major novelty is to allow VOS hardware design investigation into any application, including algorithms with dynamic behavior, which unpredictably changes the executing flow due to timing errors. Table 5.7 summarizes the related work characteristics comparing with our work. Additionally, as a case study, we investigate the SAD hardware accelerator under VOS into a motion estimation of a video encoder. In contrast to state of the art in VOS for video encoding, by employing our framework, we correctly feedback the timing errors at runtime to the encoder, resulting in a VOS investigation under a fully compliant standard.

5.5.2 Results Evaluation

This section presents the tradeoff of employing VOS in the hardware accelerator in a fully compliant encoder implementation. The method herein demonstrated allows the exploration of the boundaries of the error tolerance to maximize the energy savings per operation.

5.5.2.1 Implementation Details

The accelerator implementation of Fig. 5.3 uses industrial EDA tools for standard-cell CMOS hardware design. The closed-loop method is *tool agnostic* however, and can straightforwardly be adapted to any RTL synthesis and gate-level simulation tool-flow. The SAD hardware accelerator has parallelism to compute eight comparisons per clock cycle. The architectures were described in VHDL with the adder inferred by the logic synthesis tool (plus operator in VHDL). Cadence Genus$^{\text{TM}}$ tool performed the RTL to gate-level netlist synthesis and the STA. The hardware accelerator was mapped to 45 nm standard cells for 1.10V voltage, aiming for its maximum frequency of 1 GHz (with zero slack value). Cadence Incisive$^{\text{TM}}$ simulator performs the gate-level netlist simulations considering the gate delays. We evaluate VOS cases decreasing by 50mV in two steps, from 1.10V to 1.05V down to 1.00V, with 25, 35, 50, 75, and 100% of TGBs. All the simulations consider VOS libraries described in (AMROUCH et al., 2019) and publicly available online in (AMROUCH, 2019). The following section presents the results evaluation and the discussions for this case study implementation.

We evaluate the joint accelerator-algorithm runtime encoding the first second of the BQMall, BasketBallPass, and RaceHorses video sequences of the 416×240 resolution for four QPs, using the x265 encoder. The first second of BQMall, BasketBallPass, RaceHorses have low-, medium- and high-movement, respectively. BasketBallPass with medium-motion contains parts with low-motion and parts with high-motion. BasketBallPass with a low-motion presents the major part of the PUs with a zero-length MV (i.e., using as reference the precisely co-located PU). We ran the encoder using the superfast encoder preset with one reference frame to simulate the use in mobile systems and the highest possible CTU (64×64). QP was set according to the CTCs that recommend 22, 27, 32, and 37 to calculate the BD-BR loss. BD-BR can be directly compared in future works keeping the same: encoder, presets, videos, and QPs, which ensure comparability and reproducibility.

5.5.2.2 Coding Impacts Analysis

We quantify the ultimate encoding impact of the errors generated by the SAD hardware accelerator under VOS in terms of coding efficiency and the savings in energy per operation. The 100% absolute TGB for the SAD hardware accelerator when reducing the voltage from 1.10V to 1.05V and 1.00V is 64 ps and 146 ps, respectively. We investigate different narrowed TGBs between 0 to 100% for the three aforementioned video sequences, which contain low-, medium- and high-motion.

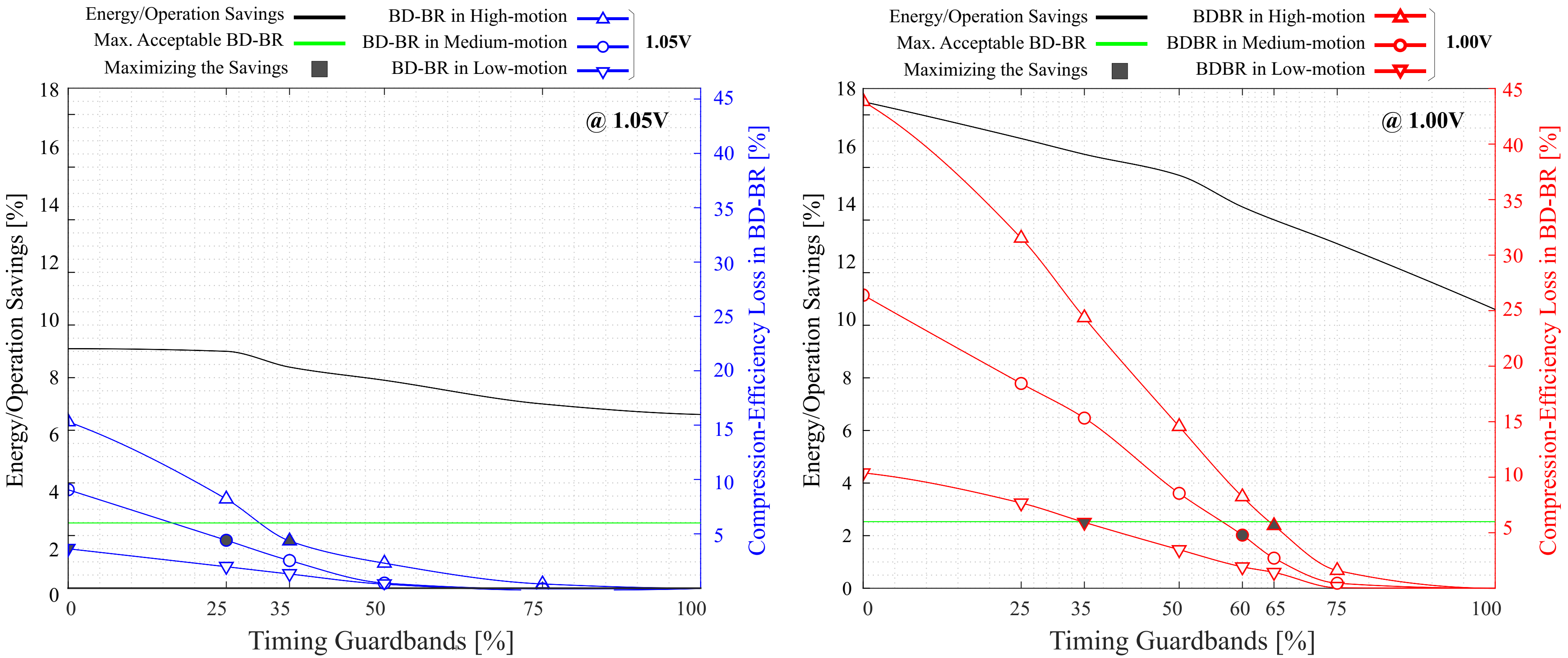
TGBs at runtime impose an operational frequency reduction and then a reduction in the frame rate. We considered 10% as a maximum operational frequency reduction allowed to employ a runtime guardband. The maximum of 10% frame rate decreases in a video of 30 fps corresponds to 27 fps, which is reasonably acceptable when operating with the maximum energy/operation savings.

Fig. 5.17 demonstrates the energy per operation savings versus coding efficiency loss (in BD-BR). The black-colored line (in Fig. 5.17) represents the savings in energy per operation calculated by P_T/F_{TGB}, where P_T is total power dissipation at the operational frequency with the guardband F_{TGB} applied. The range of energy per operation savings for 1.05V supply voltage is from 9.1% to 6.6%, and for 1.00V, it varies from 18.5% down to 10.6%. Fig. 5.17-a shows, in blue, the BD-BR impacts for 1.05V (-50mV from the nominal) and Fig. 5.17-b, in red, for 1.00V (-100mV from the nominal).

We can observe in Fig. 5.17 that the VOS in a video with low movement tends to result in a smaller video quality impact, and therefore it demands fewer TGBs. BD-BR impacts curve are in the range up to 15.6% in -50mV (at 1.05V) and 44.9% in -100mV (at 1.00V) for the higher-motion videos, which are impacted the most by VOS.

The green horizontal line (in Fig. 5.17) represents 6% of BD-BR loss as a maximum coding efficiency loss reference boundary to adopt a minimum guardband. By defining the uppermost quality boundary, we can assign to each motion quantity the required guardband for the quality and maximize the trading off in the energy per operation savings while maintaining an acceptable compression-efficiency loss. The minimum required guardband points for the maximum acceptable BD-BR boundary of 6% was gray-colored highlighted in Fig. 5.17.

Figure 5.17: Coding efficiency loss in terms of BD-BR versus the timing guardbands when reducing the voltage in -50mV (in blue) and -100mV (in red) for three different video sequences (low-, medium- and high-motion).



Source: The Author.

Table 5.8 summarizes those points with values to improve the reproducibility and comparability with future works in the literature. Table 5.8 shows the minimum required guardband TGB_{min} and the respective operational frequency with less than 6% of BD-BR of coding efficiency loss. We observe (in 5.8) that the video sequence BQSquare with low-motion behavior can be executed with -50mV (at 1.05V) of VOS without TGBs and requires only 35% of TGBs for -100mV (at 1.00V). The medium-motion video sequence (BasketBallPass) the minimum TGBs requirement increases to 25% of TGBs for -50mV (at 1.05V) and 60% to -100mV (at 1.00V).

The most sensible video sequence to VOS is with the higher-motion content, requiring at least 35% of TGB for -50mV (1.05V) and 65% for -100mV (at 1.00V). The power dissipation (in microwatts) and energy/operation (Joule per mega operations) absolute values – and its savings– were also described in Table 5.8. The range of power savings was in the range of 9.1% to 22.61%. Note that the energy/operation metric better represents the benefits since that is normalized by the operational frequency imposed by the TGB. The exact BD-BR of the best case tradeoff also was described in Table 5.8.

Table 5.8: Summary of the minimum guardband required for the SAD hardware accelerator under VOS (Fig. 5.17) constraining less than 6% of BD-BR compression efficiency ultimate impact and the respective savings in power dissipation and energy/operation.

		Voltage	$^{\alpha}$ TGB_{min}	$^{\beta}$ TGB_{min} Frequency		T. Power @ Freq. TGB_{min}		$^{\gamma}$EOP @ Freq. TGB_{min}		BD-BR
		(mV)	BD-BR < 6%	(MHz)	Reduction (%)	(μW)	Savings (%)	(J/MOP)	Savings (%)	(%)
		Nominal	0%	1000	0%	8508.0	Ref.	8.51	Ref.	0%
Video	BQSquare	-50mV	0%	1000	0%	7736.0	9.1%	7.74	9.1%	3.646%
	(Low-motion)	-100mV	35%	948.9	5.11%	6738.8	20.79%	7.10	16.5%	6.000%
	BasketBallPass	-50mV	25%	984.0	1.6%	7617	10.47%	7.74	9.0%	4.389%
	(Medium-motion)	-100mV	60%	912.4	8.76%	6631.1	22.06%	7.27	14.6%	4.746%
	RaceHorses	-50mV	35%	977.6	2.24%	7629.0	10.33%	7.80	8.3%	4.367%
	(High-motion)	-100mV	65%	905.0	9.5%	6584.5	22.61%	7.28	14.5%	5.644%

$^{\alpha}TGB_{min}$ is the runtime timing guardband to keep 6% of BD-BR.

$^{\beta}$ Frequency TGB_{min} is the operational frequency with the TGB_{min} to keep at least 6% of BD-BR.

$^{\gamma}$ EPO is Energy/Operation @ Freq. TGB_{min}, and represents the total power dissipation divided by the frequency TGB_{min}.

5.5.2.3 Prediction Residue Analysis

The result of the prediction encoding tool is the predicted block. The encoder subtracts the original block and the predicted block to generates the residue. An ideal prediction generates a zero residue block. As the quality of the motion estimation reduces the prediction efficiency in search redundancies, the residue starts to increase and the compression efficiency to decreases. Therefore, we can investigate the residue to subjectively evaluate the impact of the VOS in the prediction and the guardbanding benefits.

We generate the residual frame to perform a subjective analysis – such as the one from Figs. 5.18(d-o) – of the prediction impacts when operating under VOS. Grey-colored pixels in Fig. 5.18 means that the residue is zero. Therefore, in these pixels, the prediction stage was able to estimate the exact pixel value of the original frame.

Figs. 5.18(a-c) represent the residue with the SAD in the nominal voltage (i.e., normal operation without errors due to VOS). Firstly, we can observe in the normal operation that the residue is higher in high-motion sequences (Fig. 5.18(-a)) than medium-motion (Fig. 5.18(b)) and subsequently the sequences with lesser residue is the low-motion (Fig. 5.18(c)). As expected, the residue increases with the motion quantity video characteristics. The increase of the residue occurs due to the motion estimation task's difficulty to match a good enough when stopping the search.

The VOS impacts in the high-motion video without TGBs can be observed in Figs. 5.18(a,d,g,j,m). For both 1.05V (-50mV) and 1.00V (-100mV) without TGBs, the impact in the residue of the high-motion video is elevated. To search for a candidate that is far from the original block usually takes more effort. Therefore, high-motion sequences also tend to use a wider range of bits when accumulating worst candidates. The use of a wider range when encoding a video sequence with high-motion behavior makes the motion estimation more sensitive to errors in the MSBs caused by the timing errors. Therefore, a higher effect in the high-motion videos occurs because to perform the search for higher vectors, the SAD needs to accumulate candidates with a higher difference in the path. When the MSBs are profoundly impacted by the timing errors to confuse the search. Therefore, the guardbanding is necessary to reduce the residue, even in 1.05V (-50mV) Figs. 5.18(d). We can observe the residue reducing when comparing the Fig. 5.18(a) at a nominal voltage (without timing errors) with Figs. 5.18(g,j) under VOS without TGBs and Figs. 5.18(g,j) with VOS employing the required TGBs to keep less than 6% of BD-BR loss analyzed in the last section.

We are analyzing the residue of the medium-motion sequence in Figs. 5.18(b,e,h,k,n) we can observe that the region of the basketball player image is highly moving, and the background is mostly still. The high-movement zones are highly affected by the VOS without TGBs, which can be seen in Figs. 5.18(e,h) when comparing with the nominal voltage in 5.18(b). The guardbanding reduces the impacts in these high-movement zones that can be seen in Figs. 5.18(k,n).

Finally, the residue of the low-motion video presented in Figs. 5.18(c,f,i,l,o) results in a low impact when apply the VOS. Note The result of the prediction encoding tool is the predicted block. The encoder subtracts the original block and the predicted block to generates the residue. An ideal prediction generates a zero residue block. As the quality of the motion estimation reduces the prediction efficiency in search redundancies, the residue starts to increase and the compression efficiency to decreases. Therefore, we can investigate the residue to subjectively evaluate the impact of the VOS in the prediction and the guardbanding benefits.

We generate the residual frame to perform a subjective analysis – such as the one from Figs. 5.18(d-o) – of the prediction impacts when operating under VOS. Grey-colored pixels in Fig. 5.18 means that the residue is zero. Therefore, in these pixels, the prediction stage was able to estimate the exact pixel value of the original frame.

Figs. 5.18(a-c) represent the residue with the SAD in the nominal voltage (i.e., normal operation without errors due to VOS). Firstly, we can observe in the normal operation that the residue is higher in high-motion sequences (Fig. 5.18(-a)) than medium-motion (Fig. 5.18(b)) and subsequently the sequences with lesser residue is the low-motion (Fig. 5.18(c)). As expected, the residue increases with the motion quantity video characteristics. The increase of the residue occurs due to the motion estimation task's difficulty to match a good enough when stopping the search.

The VOS impacts in the high-motion video without TGBs can be observed in Figs. 5.18(a,d,g,j,m). For both 1.05V (-50mV) and 1.00V (-100mV) without TGBs, the impact in the residue of the high-motion video is elevated. To search for a candidate that is far from the original block usually takes more effort. Therefore, high-motion sequences also tend to use a wider range of bits when accumulating worst candidates. The use of a wider range when encoding a video sequence with high-motion behavior makes the motion estimation more sensitive to errors in the MSBs caused by the timing errors. Therefore, a higher effect in the high-motion videos occurs because to perform the search for higher vectors, the SAD needs to accumulate candidates with a higher difference in the path. When the MSBs are profoundly impacted by the timing errors to confuse the search.

Therefore, the guardbanding is necessary to reduce the residue, even in 1.05V (-50mV) Figs. 5.18(d). We can observe the residue reducing when comparing the Fig. 5.18(a) at a nominal voltage (without timing errors) with Figs. 5.18(g,j) under VOS without TGBs and Figs. 5.18(g,j) with VOS employing the required TGBs to keep less than 6% of BD-BR loss analyzed in the last section.

We are analyzing the residue of the medium-motion sequence in Figs. 5.18(b,e,h,k,n) we can observe that the region of the basketball player image is highly moving, and the background is mostly still. The high-movement zones are highly affected by the VOS without TGBs, which can be seen in Figs. 5.18(e,h) when comparing with the nominal voltage in 5.18(b). The guardbanding reduces the impacts in these high-movement zones that can be seen in Figs. 5.18(k,n).

5.5.2.4 Discussions

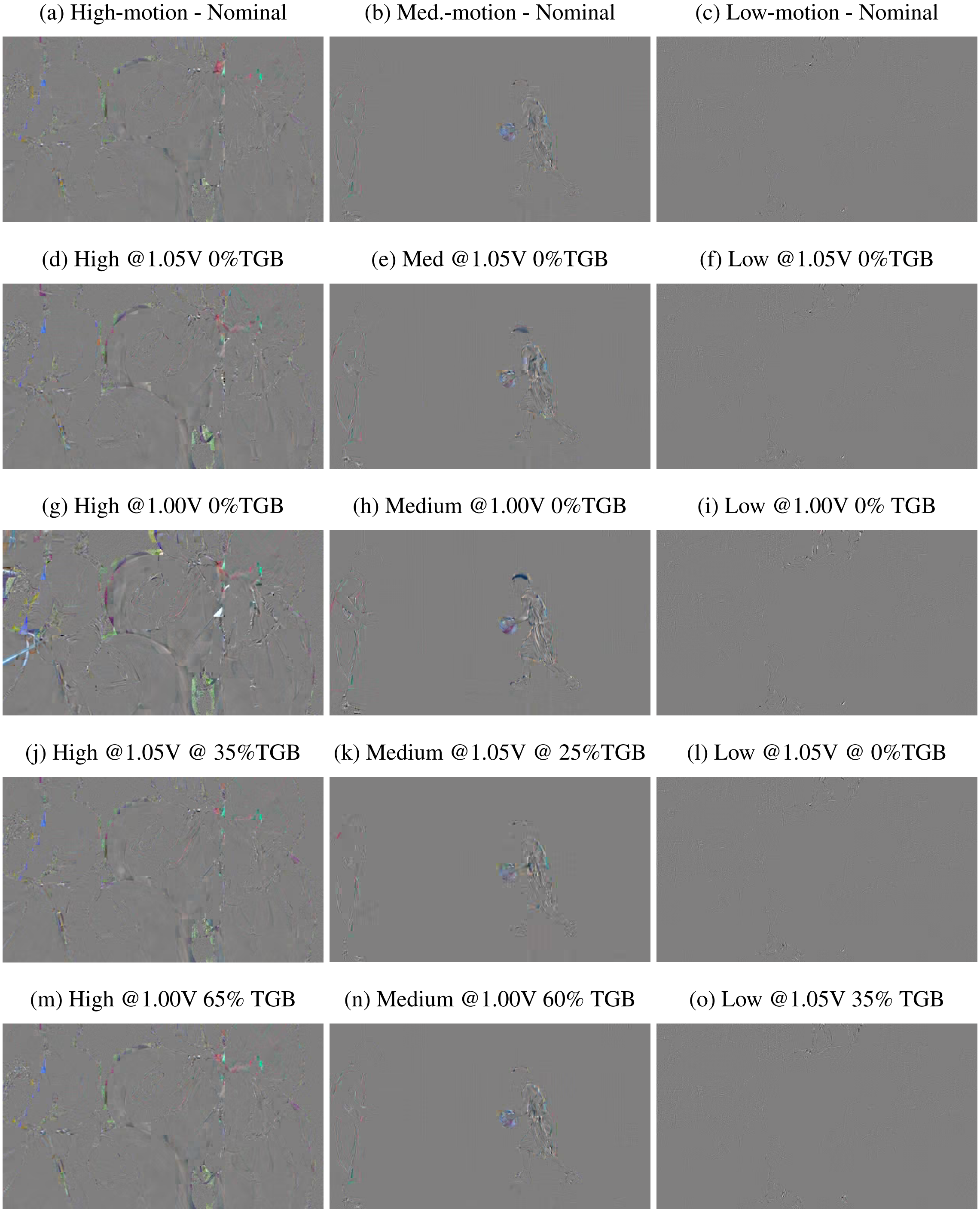
Overall, we conclude from the residue experiments that the prediction in the presence of VOS-induced timing errors without the use of narrowed TGBs can harm the video compression. From the experiments herein demonstrated, one could see clearly that, unlike the state of the art (VARATKAR; SHANBHAG, 2008) conclusions, VOS cannot be applied directly in medium and high-motion videos, even for just -50mV scaling. Therefore, the runtime narrowed TGBs herein investigated for video encoding are mandatory to maintain the coding efficiency trading-off within acceptable boundaries when operating in the VOS modes.

5.5.2.5 Comparisons with the State of the Art

Beyond and foremost, this is the first work investigating a closed-loop application with the hardware accelerators under VOS. Prior works have investigated VOS in open-loop applications, with hardware blocks relevant to their algorithms. However, in industrial-strength applications like video encoding, the timing errors must be evaluated in a closed-loop manner.

The scope of the state-of-the-art frameworks (ZERVAKIS et al., 2018; ZERVAKIS et al., 2019) is limited to examine VOS-induced errors into the output of the hardware accelerators (i.e., in the standalone operation). Their work in (ZERVAKIS et al., 2018; ZERVAKIS et al., 2019) is agnostic to the application considering that the VOS thresholds are decided in a posterior step. However, applications with a closed-loop dependency

Figure 5.18: Prediction residue: (a-c) under nominal voltage and (d-o) when the SAD hardware accelerator is running under voltage overscaling timing-errors for three test case videos (high-, medium- and low- motion).



(a) High-motion - Nominal
(b) Med.-motion - Nominal
(c) Low-motion - Nominal
(d) High @1.05V 0%TGB
(e) Med @1.05V 0%TGB
(f) Low @1.05V 0%TGB
(g) High @1.00V 0%TGB
(h) Medium @1.00V 0%TGB
(i) Low @1.00V 0% TGB
(j) High @1.05V @ 35%TGB
(k) Medium @1.05V @ 25%TGB
(l) Low @1.05V @ 0%TGB
(m) High @1.00V 65% TGB
(n) Medium @1.00V 60% TGB
(o) Low @1.05V 35% TGB

Source: The Author.

between the hardware and the algorithms (in software) to define VOS thresholds are not realistic. The evaluation of the VOS effects only at the output of the hardware accelerators in the standalone operation cannot assess the ultimate VOS impact on the application performance. In this class of applications, all the decisions are made depending on the closed-loop. Our work is the first to present and solve this issue showing how to bridge the VOS effects during the application runtime in the joint hardware accelerator-algorithm closed-loop (i.e., to the software level) . Hence, it was not possible for the state-of-the-art frameworks in (ZERVAKIS et al., 2018; ZERVAKIS et al., 2019) to define the maximum bounds of the VOS. Unlike state of the art, our framework offers a holistic approach aiming to maximize the energy savings fulfilling the good enough quality of any application (i.e., also in cases involving accelerator-algorithm closed-loops).

On the comparison on a similar case study, the work in video coding application presented in (VARATKAR; SHANBHAG, 2008) shown a standalone motion estimation accelerator under VOS investigation with noise tolerance techniques to improve the prediction. The work in (VARATKAR; SHANBHAG, 2008) analyzed the output of the motion estimation under VOS. Therefore, (VARATKAR; SHANBHAG, 2008) did not measure the ultimate impact in the compressed video (quality, bitrate, and compression efficiency). The work in (HE; GERSTLAUER; ORSHANSKY, 2013) investigate VOS in standalone DCT/IDCT accelerators. Unlike our work, the methods employed in (HE; GERSTLAUER; ORSHANSKY, 2013) cannot measure the impacts of the timing errors at the compression efficiency for the ultimate impact evaluation at the application layer. Our work herein demonstrates a framework capable of assessing the ultimate impact of closed-loop applications by online linking the hardware accelerator and the rest of the application. Our framework bridges the gap between VOS within the joint hardware accelerator-algorithm in a closed-loop (hardware/software) during the application runtime to achieve it. Additionally, we also show that in the current technology processes, the TGBs herein investigated are mandatory to maintain an acceptable compression efficiency in medium- and high-motion video sequences, even for just -50mV of VOS. The results in terms of compression efficiency shown above also enable comparing the impact in different video sequences, demonstrating that the high-motion is more susceptible to VOS than low-motion videos.

5.5.2.6 Conclusions of the Section

This work is the first to investigate the employing of VOS in the joint hardware accelerator-algorithm closed-loop of the applications with dynamic behavior. The framework was exercised by crossing VOS-induced timing errors between layers employing VOS cell libraries. We bridged the large gap between the hardware accelerator at gate-level and the algorithms dynamically during its runtime. We demonstrate our framework investigating for the first time the VOS-induced timing errors in an existing real-world full-compliant video coding application. On the VOS investigation, we revealed that at least -50mV is mandatory to employ a minimal narrowed TGB to keep the compression efficiency within an acceptable margin for medium- and high-motion video sequences. We also have shown that, when under VOS, the ultimate impact in compression efficiency is related to the motion quantity. We conclude that the higher is the motion, the more susceptible video quality is to the timing errors. Therefore, the TGBs must take into account the motion behavior of the video sequence. Further, the advantages of timing guardband controlled reduction (down at maximum 9.5% the clock frequency) for saving energy (up to 16.5% in energy/operation) during video compression are clearly quantified in the results by virtue of the developed framework.

5.6 Framework Comparisons to the State of The Art

General Usage: We herein proposed and implemented a holistic framework for crossing the errors of AxC and TS hardware design techniques underlying hardware accelerators to the algorithm and application layers. We generically demonstrated the methods employed in the framework and its full compatibility with any hardware accelerator and any application (i.e., including applications containing closed-loop algorithms). Our case studies showed investigations between the abstraction levels extremes (physical and circuit to the algorithms) to algorithms with the ultimate impact at the application to reveal the virtue of the framework. However, our framework can be potentially used for new investigations onto speculative timing designs to recover reliability across multiple and different abstraction layers (i.e., device physics, circuit, architecture, up to its algorithm interactions). For instance, at the device layer, the framework proposal recently investigates the degradation in newer technologies such as the emerging NCFET (negative capacitance field-effect transistors) (PAIM et al., 2021b). At the circuit and architecture layers, our framework can be employed to combine speculative timing design with different approximate computing techniques as (a) netlist gate-level pruning (ZERVAKIS et al., 2019), (b) approximate logic synthesis (SCARABOTTOLO et al., 2020), (c) runtime re-configurable accuracy (ZERVAKIS; AMROUCH; HENKEL, 2020). Note that the online connection in our framework enables the support for a cross-layer tuning of the aforementioned approximate techniques mentioned in (SCARABOTTOLO et al., 2020) as a promising strategy.

Potential Improvement: Our future work intends to include the support for dynamic temperature rising through online changing of the delay information (modifying the SDF at runtime). This is a new feature recently aggregated to the state-of-the-art gate-level simulators such as Cadence Xcelium (CADENCE, 2020c).

Closed-loops Support: The frameworks in (JIAO et al., 2018; MOGHADDASI; NASAB; KARGAHI, 2020) have proposed a timing error analysis less accurate than GLS employing learning methods focusing only on the functional units (i.e., combinational circuits). Then, frameworks in (JIAO et al., 2018; MOGHADDASI; NASAB; KARGAHI, 2020) cannot evaluate the timing errors in the whole hardware accelerator and its persistent internal states in time to support sequential circuits. The works in (AMROUCH et al., 2019; AFZALI-KUSHA et al., 2020) has analyzed the GLS at the circuit level, propagating, in an offline approach, the errors to the algorithm and application layers. The works in (HE; GERSTLAUER; ORSHANSKY, 2013; ZERVAKIS et al., 2018) evaluate timing errors in stand-alone video hardware accelerators. The methods employed in (HE; GERSTLAUER; ORSHANSKY, 2013; ZERVAKIS et al., 2018) cannot measure the impacts of the timing errors at the algorithm layers for the full ultimate impact evaluation at the application layer. However, many industrial-strength applications demand the timing errors impact evaluation in a closed-loop manner. Our framework solved this issue by crossing the timing errors underlying the hardware accelerators to the application layers employing degradation-aware cell libraries. Then, at open-loop implementation, they cannot capture the complex interactions between the hardware accelerators and algorithms and the ultimate impact on the application layer.

5.7 Conclusion of the Chapter

This chapter introduced a framework proposal to investigate AxC and TS hardware design techniques. The framework can dynamically cross logic and timing errors of the hardware accelerator to the algorithms and application layers. It demonstrates the complex and dynamic interaction between the algorithms and the underlying timing errors at runtime. We employ degradation-aware cell libraries for aging, temperature, and voltage over-scaling effects to achieve this goal while crossing the layers simulating the gate-level dynamically into the application. The framework proposal was herein generically demonstrated as being compatible with any application and any hardware accelerator. We evaluate the framework by investigating the effects using an existing real-world HEVC video compression application. The case studies using AxC and TS demonstrated the virtue of our holistic framework investigating the performance and energy-efficiency benefits versus the impacts on video quality and compression.

6 CONCLUSIONS OF THE THESIS

Approximate computing emerges as a design alternative to improve the energy profile of a vast number of application domains. This thesis focuses on investigating AxC and TS hardware design, connecting the impact underlying the hardware accelerators to the algorithms and application layers.

This thesis proposes a new approximation for the 8-point DTT, with a higher power- and compression-efficiency by exploring coefficient truncation, leading to the values $1/16$, $-1/16$, $1/8$, and -$1/8$. Considering operations with integers, the smaller magnitude of coefficients causes truncation in the internal transform calculations and leads to lower values for the non-diagonal residues, which reduces non-orthogonality. The results show that the approximate DTT hardware proposal increases the maximum frequency up to 64%, minimizes the circuit area in up to 43.6%, and saves up to 65.4% in power dissipation. The best DTT approximation mapped for FPGA shows an increase of up to 58.9% on the maximum clock frequency and savings of about 28.7% and 32.2% on slices and dynamic power, respectively, compared with state of the art. This approximate DTT proposal also achieves a higher compression ratio and less quality loss in the compressed image when compared to state-of-the-art approximate DTT hardware designs. Firstly, we introduce the DTT hardware architectures proposals for JPEG image compression, an application with static behavior (i.e., the errors did not change subsequent decisions).

Then, we demonstrate the challenge of considering applications that present a dynamic behavior where the closed-loop between the algorithm and hardware accelerators must be considered. This thesis introduces a framework capable of evaluating the application's ultimate impact considering the hardware accelerator produced errors (i.e., from AxC and TS hardware design). The framework proposal was herein generically demonstrated as being compatible with any application and any hardware accelerator. As a case study in this work, the hardware accelerator employed was the sum of absolute differences (SAD), the most compute-intensive accelerator on commercial video encoder for mobile applications. We evaluate the framework by investigating the AxC and TS hardware design effects using an existing real-world HEVC video compression application. We demonstrate the framework's virtue by employing three different case studies: approximate adders, aging and temperature effects, and voltage over-scaling.

The proposed method simulates the gate-level circuit dynamically inside the application, with realistic results of the impact of the adder-tree approximate logic implementation on both quality and encoder bitrate results. A comprehensive DSE is shown herein, with 13 types of 6 classes of approximate adders in the 8-way SAD accelerator hardware blocks. Over 3,000 logic variants of approximations at gate-level were developed. Our framework shows that the LOA and ETA-I approximate adders and TRUNC_0 deliver better compression-power tradeoffs. Application-level results showed that the SAD kernel based on the LOA achieve savings of up to 45% of energy/operation with an increase of only 1.9% in BD-BR.

The framework investigates temperature and aging effects across the different layers starting from transistor physics all the way up to the algorithm layer. These results introduce the ultimate impact of degradation-induced timing errors underlying hardware accelerator when interacting with the algorithms. The results demonstrate the runtime behavior impacts of three advanced block-matching algorithms of the video encoder in a joint operation by a SAD accelerator under timing errors induced by temperature and aging effects considering a 14nm FinFET technology. Our results quantify the degradation effects impact on processing overhead, compression efficiency, and video quality. Between the algorithms evaluated, the DS algorithm is the most reliable and is capable of enduring 50°C with less than 2% of BD-BR increase and, in the worst-case, 19.22% of BD-BR increase under 75°C with 10-Y aging. On the other hand, we also reveal that the UMHS is an unreliable algorithm with a 5.71% BD-BR increase under 50°C and up to 37.36% in the worst-case investigated at 75°C with 10-Y aging. The results show that the processing overhead loses the boost of about 6.96% (at 50°) to 17.41% (at 75° with 10-Y aging) in the maximum clock frequency achieved when removing the timing guardbands from 8.06% to 46.96% caused by the UMHS algorithm due to its interactions with the SAD accelerator under timing errors. Unlike the UMHS algorithm, the DS and HS algorithm's reliable behavior was revealed by our framework showing a negligible processing overhead for all cases.

We demonstrate the framework investigating VOS-induced timing errors maintaining the full compliance of the video coding application. On the VOS investigation, we revealed that at least -50mV is mandatory to employ a minimal narrowed TGB to keep the compression efficiency within an acceptable margin for medium- and high-motion video sequences. We also have shown that, when under VOS, the ultimate impact in compression efficiency is related to the motion quantity. We conclude that the higher is the motion, the more susceptible video quality is to timing errors. Therefore, the TGBs must take into account the motion behavior of the video sequence. Further, the advantages of timing guardband controlled reduction (down at maximum 9.5% the clock frequency) for saving energy (up to 16.5% in energy/operation) during video compression are clearly quantified in the results by virtue of the developed framework. The following section demonstrates examples of the next steps for future investigations.

6.1 Future Work

Approximate computing and TS hardware design investigations are in their infancy. There are many paths to conduce future works employing the framework proposal.

Towards recover reliability: The framework can fully support other future work on investigations regarding aiming to recover reliability of the TS hardware design employing approximate computing and other techniques across the multiple abstraction layers (i.e., device physics, circuit, architecture, up to its algorithm interactions). The work shown in (PAIM et al., 2021b) is an example of future work investigating an emerging device technology regarding its resiliency against the VOS effects, which boost up to 2.78x the energy efficiency of the circuit to the same error threshold.

Multiple-level Approximations: The framework can be extended and connected to system-level simulators that comprehend the compiler. Future work can employ TS hardware design combined with approximate circuits, approximate languages, compilers, and approximate storage.

Runtime Configurable AxC and TS Techniques: The framework proposal can be employed to investigate algorithms to control runtime approximations. We can extend the framework to support the control of the voltage over-scaling during the runtime.

Towards other Applications and Accelerators: The framework is generic and compatible with any accelerator and application. Therefore, future works aim also to comprehend applications of machine learning and digital signal processing. Video coding accelerators are mostly based on adders. Therefore, other applications also enable the exploration of other arithmetic operators as multipliers and divisors.

REFERENCES

ABREU, B. et al. Exploiting Absolute Arithmetic for Power-Efficient Sum of Absolute Differences. **IEEE International Conference on Electronics, Circuits and Systems (ICECS)**, Batumi, Georgia, 2017.

AFZALI-KUSHA, H.; KAMAL, M.; PEDRAM, M. Low-power Accuracy-configurable Carry Look-ahead Adder Based on Voltage Overscaling Technique. In: **21st International Symposium on Quality Electronic Design (ISQED)**. [S.l.: s.n.], 2020. p. 67–72.

AFZALI-KUSHA, H. et al. Design Exploration of Energy-Efficient Accuracy-Configurable Dadda Multipliers With Improved Lifetime Based on Voltage Overscaling. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 28, n. 5, p. 1207–1220, 2020.

AGRAWAL, A. et al. Approximate computing: Challenges and opportunities. In: **IEEE International Conference on Rebooting Computing (ICRC)**. [S.l.: s.n.], 2016. p. 1–8.

ALAN, T.; HENKEL, J. SlackHammer: Logic Synthesis for Graceful Errors Under Frequency Scaling. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 37, n. 11, p. 2802–2811, Nov 2018.

ALBICOCCO, P. et al. Imprecise Arithmetic for Low Power Image Processing. In: **Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)**. [S.l.: s.n.], 2012. p. 983–987.

AMROUCH, H. **Degradation-Aware Cell Libraries**. 2019. Http://ces.itec.kit.edu/dependable-hardware.php.

AMROUCH, H. et al. On the Efficiency of Voltage Overscaling under Temperature and Aging Effects. **IEEE Transactions on Computers**, p. 1647–1662, May 2019.

ASSARE, O.; GUPTA, R. Accurate Estimation of Program Error Rate for Timing-Speculative Processors. In: **2019 56th ACM/IEEE Design Automation Conference (DAC)**. [S.l.: s.n.], 2019. p. 1–6.

BATEMAN, H. et al. **Higher Transcendental Functions**. [S.l.]: McGraw-Hill, 1953.

BAYER, F.; CINTRA, R. Image Compression via a Fast DCT Approximation. **IEEE Latin America Transactions**, v. 8, n. 6, p. 708–713, Dec 2010.

BJONTEGAARD, G. Calcuation of average PSNR differences between RD-curves. In: **Doc. VCEG-M33 ITU-T Q6/16**. [S.l.: s.n.], 2001.

Bohr, M. T. Logic Technology Scaling to Continue Moore's Law. In: **2018 IEEE 2nd Electron Devices Technology and Manufacturing Conference (EDTM)**. [S.l.: s.n.], 2018. p. 1–3.

BOSIO, A.; MENARD, D.; SENTIEYS, O. A Comprehensive Analysis of Approximate Computing Techniques: From Component- to Application-Level. In: **Embedded Systems Week (ESWEEK)**. [S.l.: s.n.], 2018. p. 1–2.

BOSSEN, F. et al. Common test conditions and software reference configurations. **JCTVC-L1100**, v. 12, 2013.

BRKIC, S.; IVANIš, P.; VASIć, B. Majority Logic Decoding Under Data-Dependent Logic Gate Failures. **IEEE Transactions on Information Theory**, v. 63, n. 10, p. 6295–6306, 2017.

BRKIC, S.; IVANIS, P.; VASIć, B. Hard-decision decoding of LDPC codes under timing errors: Overview and new results. In: **25th Telecommunication Forum (TELFOR**. [S.l.: s.n.], 2017. p. 1–8.

BUDAGAVI, M. et al. Core Transform Design in the High Efficiency Video Coding (HEVC) Standard. **IEEE Journal of Selected Topics in Signal Processing**, v. 7, n. 6, p. 1029–1041, Dec 2013.

CADENCE. **Cadence EDA tools**. 2020. Http://www.cadence.com.

CADENCE. **Manual: Genus Synthesis Flow**. 2020. Http://www.cadence.com.

CADENCE. **Palladium Z1 Enterprise Platform and Xcelium Logic Simulator**. 2020. Https://www.cadence.com.

CHANDRAKASAN, A. P.; BRODERSEN, R. W. **Low Power Digital CMOS Design**. Springer US, 1995. Available from Internet: <https://doi.org/10.1007/978-1-4615-2325-3>.

CHEN, J.; HU, J. Energy-Efficient Digital Signal Processing via Voltage-Overscaling-Based Residue Number System. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 21, n. 7, p. 1322–1332, 2013.

CHEN, M. et al. An error-resilient wavelet-based ECG processor under voltage overscaling. In: **IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings**. [S.l.: s.n.], 2014. p. 628–631. ISSN 2163-4025.

CHENG, E. et al. Tolerating Soft Errors in Processor Cores Using CLEAR (Cross-Layer Exploration for Architecting Resilience). **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 37, n. 9, p. 1839–1852, 2018.

CHIPPA, V. K. et al. Analysis and characterization of inherent application resilience for approximate computing. In: **50th ACM/EDAC/IEEE Design Automation Conference (DAC)**. [S.l.: s.n.], 2013. p. 1–9.

CHOUDHURY, M. R. et al. Time-Borrowing Circuit Designs and Hardware Prototyping for Timing Error Resilience. **IEEE Transactions on Computers**, v. 63, n. 2, p. 497–509, 2014.

CINTRA, R.; BAYER, F. A DCT Approximation for Image Compression. **IEEE Signal Processing Letters**, v. 18, n. 10, p. 579–582, Oct 2011.

CISCO. **Cisco Annual Internet Report (2018–2023) White Paper**. 2018. Https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf.

CLARK, L. T. et al. ASAP7: A 7-nm finFET predictive process design kit. **Microelectronics Journal**, v. 53, 2016.

CONCEIÇÃO, R. et al. Low-Cost and High-Throughput Hardware Design for the HEVC 16x16 2-D DCT Transform. **Journal of Integrated Circuits and Systems**, v. 9, n. 1, p. 25–35, Dec 2014.

Dennard, R. H. Past Progress and Future Challenges in LSI Technology: From DRAM and Scaling to Ultra-Low-Power CMOS. **IEEE Solid-State Circuits Magazine**, v. 7, n. 2, p. 29–38, 2015.

DO, T. T. T.; TAN, Y. H.; YEO, C. High-throughput and low-cost hardware-oriented integer transforms for HEVC. In: **IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2014. p. 2105–2109.

DUARTE, J. P. et al. BSIM-CMG: Standard FinFET compact model for advanced circuit design. In: **ESSCIRC Conference 2015 - 41st European Solid-State Circuits Conference (ESSCIRC)**. [S.l.: s.n.], 2015. p. 196–201.

DUTT, S.; NANDI, S.; TRIVEDI, G. Analysis and Design of Adders for Approximate Computing. **ACM Transactions on Embedded Computing Systems (TECS)**, v. 17, n. 40, p. 40:1–40:28, Dec 2017.

EBRAHIMI, M. et al. Aging-aware logic synthesis. In: **IEEE/ACM International Conference on Computer-Aided Design (ICCAD)**. [S.l.: s.n.], 2013. p. 61–68.

EL-HAROUNI, W. et al. Embracing Approximate Computing for Energy-Efficient Motion Estimation in High Efficiency Video Coding. In: **Design, Automation & Test in Europe Conference & Exhibition (DATE)**. [S.l.: s.n.], 2017. p. 1384–1389. ISSN 1522-4880.

ENOMOTO, T.; KOBAYASHI, N. A low power multimedia processor implementing dynamic voltage and frequency scaling technique. In: **2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)**. [S.l.: s.n.], 2013. p. 75–76. ISSN 2153-6961.

ESMAEILZADEH, H. et al. Architecture support for disciplined approximate programming. In: **Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems**. [S.l.: s.n.], 2012. p. 301–312.

GRELLERT, M. et al. An adaptive workload management scheme for hevc encoding. In: **IEEE International Conference on Image Processing**. [S.l.: s.n.], 2013. p. 1850–1854. ISSN 1522-4880.

GUPTA, V. et al. Low-Power Digital Signal Processing Using Approximate Adders. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 32, n. 1, p. 124–137, Jan 2013. ISSN 0098-3063.

HAN, J.; ORSHANSKY, M. Approximate Computing: An emerging paradigm for energy-efficient design. In: **2013 18th IEEE European Test Symposium (ETS)**. [S.l.: s.n.], 2013. p. 1–6.

HAN, J. et al. An Area-Efficient Error-Resilient Ultralow-Power Subthreshold ECG Processor. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 63, n. 10, p. 984–988, 2016.

HE, K.; GERSTLAUER, A.; ORSHANSKY, M. Circuit-Level Timing-Error Acceptance for Design of Energy-Efficient DCT/IDCT-Based Systems. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 23, n. 6, p. 961–974, 2013.

HEVC Test Model (HM) v. 16.7. 2016. Http://HEVC.hhi.fraunhofer.de.

HUANG, J.; LACH, J.; ROBINS, G. A Methodology for Energy-Quality Tradeoff Using Imprecise Hardware. In: **ACM/IEEE Design Automation Conference (DAC) 2012**. [S.l.: s.n.], 2012. p. 504–509. ISSN 1522-4880.

ISHIDA, R.; SATO, T.; UKEZONO, T. Approximate Adder Generation for Image Processing Using Convolutional Neural Network. In: **International SoC Design Conference (ISOCC)**. [S.l.: s.n.], 2018. p. 38–39.

ISHWAR, S.; MEHER, P. K.; SWAMY, M. N. S. Discrete Tchebichef Transform-A fast 4x4 algorithm and its application in image/video compression. In: **IEEE International Symposium on Circuits and Systems (ISCAS), 2008**. [S.l.: s.n.], 2008. p. 260–263.

ITU-T. **BD-BR Calculation ITU-T Meeting, Austin, Texas.** 2001. Http://wftp3.itu.int/av-arch/video-site/0104_Aus/VCEG-M34.xls.

ITU-T; ISO/IEC. Advanced video coding for generic audiovisual services. In: . [S.l.: s.n.], 2011.

ITU-T; ISO/IEC. High Efficiency Video Coding. In: . [S.l.: s.n.], 2013.

JEON, D. et al. Design Methodology for Voltage-Overscaled Ultra-Low-Power Systems. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 59, n. 12, p. 952–956, 2012.

JIAO, X. et al. Combining structural and timing errors in overclocked inexact speculative adders. In: **Design, Automation Test in Europe Conference Exhibition (DATE), 2017**. [S.l.: s.n.], 2017. p. 482–487.

JIAO, X. et al. CLIM: A Cross-Level Workload-Aware Timing Error Prediction Model for Functional Units. **IEEE Transactions on Computers**, v. 67, n. 6, p. 771–783, 2018.

KAMMOUN, A. et al. Forward-Inverse 2D Hardware Implementation of Approximate Transform Core for the VVC Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 30, n. 11, p. 4340–4354, 2020.

KEANE, J.; KIM, C. H. Transistor aging. **IEEE *Spectrum***, 2011.

KHONGSIT, R.; RANGABABU, P. Scalable discrete Tchebichef Transform for image/video compression. In: **2017 International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)**. [S.l.: s.n.], 2017. p. 127–131.

KOUADRIA, N. et al. Pruned discrete T chebichef transform for image coding in wireless multimedia sensor networks. **AEU - International Journal of Electronics and Communications**, v. 74, p. 123 – 127, 2017.

LIU, C.; HAN, J.; LOMBARDI, F. An analytical framework for evaluating the error characteristics of approximate adders. **IEEE Transactions on Computers**, v. 64, n. 5, p. 1268–1281, May 2015.

LIU, R.; PARHI, K. K. Power Reduction in Frequency-Selective FIR Filters Under Voltage Overscaling. **IEEE Journal on Emerging and Selected Topics in Circuits and Systems**, v. 1, n. 3, p. 343–356, Sep. 2011. ISSN 2156-3365.

LIU, Y.; ZHANG, T.; PARHI, K. K. Computation Error Analysis in Digital Signal Processing Systems With Overscaled Supply Voltage. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 18, n. 4, p. 517–526, 2010.

MAHDIANI, H. et al. Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 57, n. 4, p. 850–862, Apr 2010.

MARTINA, M. (Ed.). **VLSI Architectures for Future Video Coding**. Institution of Engineering and Technology (IET), 2019. (Materials, Circuits; Devices). Available from Internet: <https://digital-library.theiet.org/content/books/cs/pbcs053e>.

MASERA, M.; MARTINA, M.; MASERA, G. Adaptive Approximated DCT Architectures for HEVC. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 27, n. 12, p. 2714–2725, 2017.

MCCANN, K. et al. **HM10: High Efficiency Video Coding Test Model (HM10) Encoder Description**. Geneva, Switzerland, 2013.

MEHER, P. K. et al. Efficient Integer DCT Architectures for HEVC. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 24, n. 1, p. 168–178, Jan 2014.

MISHRA, S. et al. A Simulation Study of NBTI Impact on 14-nm Node FinFET Technology for Logic Applications: Device Degradation to Circuit-Level Interaction. **IEEE Transactions on Electron Devices**, v. 66, n. 1, p. 271–278, Jan 2019.

MITTAL, S. A Survey of Techniques for Approximate Computing. **ACM Computing Surveys (CSUR)**, v. 48, n. 4, p. 1–33, May 2016.

MOGHADDASI, I.; NASAB, M. E. S.; KARGAHI, M. Aging-Aware Instruction-Level Statistical Dynamic Timing Analysis for Embedded Processors. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 28, n. 2, p. 433–442, 2020.

MOHANTY, B. K. Parallel VLSI Architecture for Approximate Computation of Discrete Hadamard Transform. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 30, n. 12, p. 4944–4952, 2020.

MYERS, J. et al. A Subthreshold ARM Cortex-M0+ Subsystem in 65 nm CMOS for WSN Applications with 14 Power Domains, 10T SRAM, and Integrated Voltage Regulator. **IEEE Journal of Solid-State Circuits**, v. 51, n. 1, p. 31–44, 2016.

NAKAGAKI, K.; MUKUNDAN, R. A Fast 4 x 4 Forward Discrete Tchebichef Transform Algorithm. **IEEE Signal Processing Letters**, v. 14, n. 10, p. 684–687, Oct 2007.

NGUYEN, T. et al. Transform coding techniques in hevc. **IEEE Journal of Selected Topics in Signal Processing**, v. 7, n. 6, p. 978–989, Dec 2013. ISSN 1932-4553.

OLIVEIRA, P. et al. Low-complexity Image and Video Coding Based on an Approximate Discrete Tchebichef Transform. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 27, n. 5, p. 1066 – 1076, May 2017.

OLIVEIRA, P. A. et al. A Discrete Tchebichef Transform Approximation for Image and Video Coding. **IEEE Signal Processing Letters**, v. 22, n. 8, p. 1137–1141, Aug 2015.

PAIM, G. et al. Bridging the Gap Between Voltage Over-Scaling and Joint Hardware Accelerator-Algorithm Closed-Loop. **IEEE Transactions on Circuits and Systems for Video Technology**, p. 1–1, 2021.

PAIM, G. et al. On the Resiliency of NCFET Circuits against Voltage Over-Scaling. **IEEE Transactions on Circuits and Systems I: Regular Papers**, p. 1–1, 2021.

PAIM, G. et al. A Framework for Crossing Temperature-Induced Timing Errors Underlying Hardware Accelerators to the Algorithm and Application Layers. **IEEE Transactions on Computers**, p. 1–1, 2021.

PAIM, G. et al. A Cross-Layer Gate-Level-to-Application Co-Simulation for Design Space Exploration of Approximate Circuits in HEVC Video Encoders. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 30, n. 10, p. 3814–3828, 2020.

PAIM, G. et al. Power-, Area-, and Compression-Efficient Eight-Point Approximate 2-D Discrete Tchebichef Transform Hardware Design Combining Truncation Pruning and Efficient Transposition Buffers. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 66, n. 2, p. 680–693, 2019.

PAIM, G. et al. Pruning and approximation of coefficients for power-efficient 2-D Discrete Tchebichef Transform. In: **15th IEEE International New Circuits and Systems Conference (NEWCAS)**. [S.l.: s.n.], 2017. p. 25–28.

PALTRINIERI, A. et al. Approximate-Computing Architectures for Motion Estimation in HEVC. In: **2nd New Generation of Circuits & Systems Conference (NGCAS)**. [S.l.: s.n.], 2018. p. 190–193.

PASHAEIFAR, M. et al. A Theoretical Framework for Quality Estimation and Optimization of DSP Applications Using Low-Power Approximate Adders. **IEEE Transactions on Circuits and Systems-I: Regular Papers**, v. 66, n. 1, p. 327–340, Jan 2019.

PORTO, R. et al. Energy-efficient motion estimation with approximate arithmetic. In: **2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)**. [S.l.: s.n.], 2017. p. 1–6. ISSN 1522-4880.

PORTO, R. et al. Power-Efficient Approximate SAD Architecture with LOA Imprecise Adders. In: **IEEE 10th Latin American Symposium on Circuits Systems (LASCAS)**. [S.l.: s.n.], 2019. p. 65–68.

PORTO, R. E. C. **Desenvolvimento Arquitetural para Estimação de Movimento de Blocos de Tamanhos Variáveis Segundo o Padrao H.264/AVC de Compressão de Vídeo Digital**. Dissertation (Dissertação de Mestrado) — PPGC/UFRGS, Porto Alegre/RS, 2008a.

POTLURI, U. S. et al. Improved 8-Point Approximate DCT for Image and Video Compression Requiring Only 14 Additions. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 61, n. 6, June 2014.

PRABAKARAN, B. et al. DeMAS: An efficient design methodology for building approximate adders for FPGA-based systems. In: **Design, Automation & Test in Europe Conference & Exhibition (DATE)**. [S.l.: s.n.], 2018. p. 917–920.

PRABAKARAN, B. S. et al. Approximate Multi-Accelerator Tiled Architecture for Energy-Efficient Motion Estimation. In: ____. **Approximate Circuits: Methodologies and CAD**. [S.l.]: Springer International Publishing, 2019. p. 249–268. ISBN 978-3-319-99322-5.

PRATTIPATI, S. et al. A fast 8x8 Integer Tchebichef Transform and comparison with integer cosine transform for image compression. In: **IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS), 2013**. [S.l.: s.n.], 2013. p. 1294–1297.

PRATTIPATI, S.; SWAMY, M. N. S.; MEHER, P. K. A variable quantization technique for image compression using integer Tchebichef transform. In: **2013 9th International Conference on Information, Communications Signal Processing**. [S.l.: s.n.], 2013. p. 1–5.

RAHMALAN, H.; ABU, N. A.; WONG, S. L. Using Tchebichef moment for fast and efficient image compression. **Pattern Recognition and Image Analysis**, v. 20, n. 4, p. 505–512, 2010. ISSN 1555-6212.

SALOMON, D.; MOTTA, G. **Handbook of data compression**. [S.l.]: Springer Science & Business Media, 2010.

SCARABOTTOLO, I. et al. Approximate Logic Synthesis: A Survey. **Proceedings of the IEEE**, p. 1–19, 2020.

SCHLACHTER, J. et al. Design and applications of approximate circuits by gate-level pruning. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, IEEE, v. 25, n. 5, p. 1694–1702, 2017.

SEDIGHI, B.; ANTHAPADMANABHAN, N. P.; SUVAKOVIC, D. Timing errors in LDPC decoding computations with overscaled supply voltage. In: **IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)**. [S.l.: s.n.], 2014. p. 201–206.

SELVO, P. et al. An Optimized Partial-Distortion-Elimination Based Sum-of-Absolute-Differences Architecture for High-Efficiency-Video-Coding. In: **The 6th Conference on Applications in Electronics Pervading Industry, Environment and Society (ApplePies)**. [S.l.: s.n.], 2018. p. 1–6.

SENAPATI, U. C. P. R. K.; MAHAPATRA, K. K. Reduced Memory, Low Complexity Embedded Image Compression Algorithm Using Hierarchical Listless Discrete Tchebichef Transform. **IET Image Processing**, v. 8, p. 213–238, 2014.

SERGE. **Bjontegaard2 function – MATLAB Central File Exchange**. 2013. Https://www.mathworks.com/matlabcentral/fileexchange/41751-verification-test-for-bjontegaard2-function.

SHAFIQUE, M. et al. A low latency generic accuracy configurable adder. In: **52nd ACM/EDAC/IEEE Design Automation Conference (DAC)**. [S.l.: s.n.], 2015. p. 1–6.

SHAFIQUE, M. et al. Invited: Cross-layer approximate computing: From logic to architectures. In: **53nd ACM/EDAC/IEEE Design Automation Conference (DAC)**. [S.l.: s.n.], 2016. p. 1–6.

SILVACO. **Silvaco 15 nm Open Cell Library**. 2019. Http://www.si2.org/open-cell-library/.

SILVEIRA, B. et al. Power-Efficient Sum of Absolute Differences Hardware Architecture Using Adder Compressors for Integer Motion Estimation Design. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 64, n. 12, p. 3126–3137, Dec 2017. ISSN 1549-8328.

SOARES, L.; COSTA, E.; BAMPI, S. Design of area and energy-efficient digital CMOS FIR filters with approximate adder circuits. **Analog Integrated Circuits and Signal Processing**, v. 89, p. 99–109, Sep 2016.

SOARES, L. B. et al. Design Methodology to Explore Hybrid Approximate Adders for Energy-Efficient Image and Video Processing Accelerators. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 66, n. 6, p. 2137–2150, June 2019. ISSN 1558-0806.

ST. **ST 65nm Standard Cell Library**. 2013. Www.st.com.

STANLEY-MARBELL, P. et al. Exploiting Errors for Efficiency: A Survey from Circuits to Applications. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 53, n. 3, jun 2020.

SULLIVAN, G. J. et al. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Transactions Circuits Systems Video Technology**, v. 22, n. 12, p. 1649–1668, Dec 2012. ISSN 1051-8215.

TU, F. et al. Reconfigurable Architecture for Neural Approximation in Multimedia Computing. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 29, n. 3, p. 892–906, 2019.

USC-SIPI. **The USC-SIPI image database**. 2017. Http://sipi.usc.edu/database/.

VALLERO, A. et al. SyRA: Early System Reliability Analysis for Cross-Layer Soft Errors Resilience in Memory Arrays of Microprocessor Systems. **IEEE Transactions on Computers**, v. 68, n. 5, p. 765–783, 2019.

VARATKAR, G. V.; SHANBHAG, N. R. Energy-efficient Motion Estimation using Error-Tolerance. In: **ISLPED'06 Proceedings of the 2006 International Symposium on Low Power Electronics and Design**. [S.l.: s.n.], 2006. p. 113–118.

VARATKAR, G. V.; SHANBHAG, N. R. Error-Resilient Motion Estimation Architecture. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 16, n. 10, p. 1399–1412, Oct 2008. ISSN 1557-9999.

VERMA, K.; BRISK, P.; IENNE, P. Variable latency speculative addition: A new paradigm for arithmetic circuit design. In: **2008 Design, Automation & Test in Europe (DATE)**. [S.l.: s.n.], 2008. p. 1250–1255. ISSN 1522-4880.

WANG, S. et al. VLSI Implementation of HEVC Motion Compensation with Distance Biased Direct Cache Mapping for 8K UHDTV Applications. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 27, n. 2, p. 380–393, Feb 2017. ISSN 1051-8215.

WANG, X.; ROBINSON, W. H. Error Estimation and Error Reduction With Input-Vector Profiling for Timing Speculation in Digital Circuits. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 38, n. 2, p. 385–389, 2019.

WEIßBRICH, M. et al. FLINT+: A runtime-configurable emulation-based stochastic timing analysis framework. **Integration**, v. 69, p. 120 – 137, 2019. ISSN 0167-9260.

WESTE, N.; ESHRAGHIAN, K. **Principles of CMOS VLSI Design: A Systems Perspective**. [S.l.]: Addison-Wesley, 1994. (VLSI systems series). ISBN 9780321223371.

WU, Y. et al. An Efficient Method for Calculating the Error Statistics of Block-Based Approximate Adders. **IEEE Transactions on Computers**, v. 68, n. 1, p. 21–38, Jan 2019.

XILINX. **Manual: Vivado Design Suite**. 2019. Http://www.xilinx.com.

XU, Q.; MYTKOWICZ, T.; KIM, N. Approximate Computing: A Survey. **IEEE Design & Test**, v. 33, n. 1, p. 8–22, Feb 2016.

ZERVAKIS, G.; AMROUCH, H.; HENKEL, J. Design Automation of Approximate Circuits with Runtime Reconfigurable Accuracy. **IEEE Access**, v. 8, n. 1, p. 53522–53538, 2020.

ZERVAKIS, G. et al. VADER: Voltage-Driven Netlist Pruning for Cross-Layer Approximate Arithmetic Circuits. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 27, n. 6, p. 1460–1464, 2019.

ZERVAKIS, G. et al. VOSsim: A Framework for Enabling Fast Voltage Overscaling Simulation for Approximate Computing Circuits. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 26, n. 6, p. 1204–1208, June 2018.

ZERVAKIS, G. et al. Multi-Level Approximate Accelerator Synthesis Under Voltage Island Constraints. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 66, n. 4, p. 607–611, 2019.

ZHANG, J. et al. ThUnderVolt: Enabling Aggressive Voltage Underscaling and Timing Error Resilience for Energy Efficient Deep Learning Accelerators. In: **2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)**. [S.l.: s.n.], 2018. p. 1–6.

ZHANG, Z. et al. Optimal slope ranking: An approximate computing approach for circuit pruning. In: IEEE. **2018 IEEE International Symposium on Circuits and Systems (ISCAS)**. Florence, 2018. p. 1–4.

ZHU, N. et al. Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 18, n. 8, p. 1225–1229, Ago 2010.

ZHU, N.; GOH, W. L.; YEO, K. S. An enhanced low-power high-speed Adder For Error-Tolerant application. In: **Proceedings of the 12th International Symposium on Integrated Circuits**. [S.l.: s.n.], 2009. p. 69–72. ISSN 2325-0631.

ANNEX A

Summary of the research achievements and publications:

- A sandwich Ph.D. scholarship approved, funded by the CNPq from June-2019 to March-2020.

- One book chapter, about low-power design techniques, was published in an IET book entitled: VLSI Architectures for future Video Coding.

- One IEEE TC, two IEEE TCSVT, and two IEEE TCAS-I journals directly addressing this thesis were published.

- 24 journal papers were submitted/accepted during this Ph.D.A total of 10 of those were published in IEEE Transactions journals.

- 27 conference papers were accepted during the Ph.D. (2016-2021).

- CAPES/PROBRAL (2020-2022) bilateral funding project approval for international collaborative research with an important German research institute - Karlsruher Institute für Technology (KIT) (KIT). The theme and focus of the project are closely related to this Ph.D. thesis research.

- CAPES/FCT (2020-2021) bilateral funding project approval for an international collaboration research with an important Portuguese research institute, Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento em Lisboa (INESD-ID), in which the central themes of the project are also directly related to this Ph.D. thesis.

Table 6.1: Book Chapter.

#	O	Chapter Title	Book Title	Editorial (Country)	Y
1	1	Low-power Circuit Design Techniques for High-Resolution Video Coding	VLSI Architectures for future Video Coding	The IET (England)	2019

Table 6.2: All journal papers produced during this PhD. The titles which are directly related to this thesis are highlited in bold.

#	O	Title	Journal	Q	Year
1	1^0	**Power- Area- and Compression-Efficient 8-point Approximate 2-D Discrete Tchebichef Transform Hardware Design Combining Truncation Pruning and Efficient Transposition Buffers**	IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS-I) (Published)	A1	2019
2	1^0	**Cross-layer Gate-Level-to-Application Co-simulation for Design Space Exploration of Approximate Circuits in HEVC Video Encoders**	IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) (Published)	A1	2020
3	1^0	**Bridging the Gap Between Voltage Over-Scaling and Joint Hardware Accelerator-Algorithm Closed-Loop**	IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) (Accepted)	A1	2021
4	1^0	**A Framework for Crossing Temperature-Induced Timing Errors Underlying Hardware Accelerators to the Algorithm and Application Layers**	IEEE Transactions on Computers (TC) (Accepted)	A1	2021
5	1^0	**On the Resiliency of NCFET Circuits against Voltage Over-Scaling**	IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS-I) (Accepted)	A1	2021
6	3^0	Approximate Pruned and Truncated Haar Discrete Wavelet Transform VLSI Hardware for Energy-Efficient ECG Signal Processing	IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS-I) (Accepted)	A1	2021
7	3^0	An Energy-Efficient Haar Wavelet Transform Architecture for Respiratory Signal Processing	IEEE Transactions on Circuits and Systems II: Express Briefs (TCAS-II) (Accepted)	A1	2020
8	2^0	Power-Efficient Sum of Absolute Differences Hardware Architecture Using Adder Compressors for Integer Motion Estimation Design	IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS-I) (Published)	A1	2017
9	2^0	Architectural Exploration for Energy-Efficient Fixed-Point Kalman Filter VLSI Design	IEEE Transactions on Very Large Scale Integration (VLSI) Systems (in Minor Review)	A2	2021
10	3^0	TT-WDDL: A Triple Track Wave Dynamic Differential Logic to Maximize the Safety Against Power Attacks with Standard Cell VLSI Design	IEEE Transactions on Circuits and Systems II: Express Briefs (TCAS-II) (Submitted)	A1	2021
11	2^0	Exploring High-Order Adder Compressors for Reducing Power in Sum of Absolute Differences Architectures for UHD Video Encoding	Journal of Real-time Image Processing - Springer (JRTIP) (Published)	A2	2020
12	2^0	Power-Efficient Approximate Newton-Raphson Integer Divider Applied to NLMS Adaptive Filter for High-Quality Interference Cancelling	Circuits, Systems and Signal Processing - Springer (CSSP) (Published)	A2	2020
13	2^0	The 4-2 Fused Adder-Subtractor Compressor for Low-power Butterfly-based Hardware Architectures	Circuits, Systems and Signal Processing - Springer (CSSP) (Submitted)	A2	2020
14	2^0	Exploring NLMS-based Adaptive Filter Hardware Architectures for Eliminating Power Line Interference in EEG Signals	Circuits, Systems and Signal Processing - Springer (CSSP) (Accepted))	A2	2020
15	2^0	Low-power Fast Fourier Transform Hardware Architecture Combining a Split-Radix Butterfly and Efficient Adder Compressors	IET Computers & Digital Techniques (Accepted)	A4	2020
16	1^0	Exploring Multi-Level Composition and Efficient MCM Schemes for an Energy-Efficient Wavelet Haar Architecture	Journal of Integrated Circuits and Systems (JICS) (Submitted)	A4	2020
17	1^0	Exploring the CORDIC Algorithm and Clock-Gating for Power-Efficient Fast Fourier Transform Hardware Architectures	Journal of Integrated Circuits and Systems (JICS) (Submitted)	A4	2020
18	1^0	Framework-based Arithmetic Datapath Generation to Explore Parallel Binary Multipliers	Journal of Integrated Circuits and Systems (JICS) (Published)	A4	2020
19	1^0	Pruned Discrete Tchebichef Transform Approximation for Low-power Hardware Design	Journal of Integrated Circuits and Systems (JICS) (Published)	A4	2018
20	2^0	A Low-Area and Fast 8-2 Adder Compressor Monolithic Circuit	Journal of Integrated Circuits and Systems (JICS) (Published)	A4	2019
21	3^0	Enhancing Side Channel Attack-Resistance of the STTL Combining Multi-$\mathrm{V_t}$ Transistors with Capacitance and Current Paths Counterbalancing	Journal of Integrated Circuits and Systems (JICS) (Published)	A4	2019
22	2^0	Exploring Absolute Difference Arithmetic for Power-Efficient Sum of Absolute Differences	Journal of Integrated Circuits and Systems (JICS) (Published)	A4	2020
23	1^0	Exploring approximations in 4- and 8- point DTT hardware architectures for low-power image compression	Analog Integrated Circuits and Signal Processing (Published)	B1	2018
24	2^0	Novel Hybrid Encoding Arithmetic Operators for Energy-Efficient HEVC Quantization	Integration, the VLSI Journal (Minor Review)	A4	2020

Table 6.3: Published International Conference Papers (from 2019 up to 2021).

#	O	Title	Conference	Year
1	2^0	Exploring Approximate Adders for Power-Efficient Harmonics Elimination Hardware Architectures	IEEE Latin American Symposium on Circuits and Systems (LASCAS'21)	2021
2	2^0	A Power-Efficient FFT Hardware Architecture Exploiting Approximate Adders	IEEE Latin American Symposium on Circuits and Systems (LASCAS'21)	2021
3	2^0	Exploring NLMS and IPNLMS Adaptive Filtering VLSI Hardware Architectures for Robust EEG Signal Artifacts Elimination	IEEE International Conference on Electronics, Circuits and Systems (ICECS'20)	2020
4	2^0	Exploring Efficient Adder Compressors for Power-Efficient Sum of Squared Differences Design	IEEE International Conference on Electronics, Circuits and Systems (ICECS'20)	2020
5	2^0	The Radix-2^m Squared Multiplier	IEEE International Conference on Electronics, Circuits and Systems (ICECS'20)	2020
6	2^0	An Efficient NLMS-based VLSI Architecture for Robust FECG Extraction and FHR Processing	IEEE International Conference on Electronics, Circuits and Systems (ICECS'20)	2020
7	2^0	Optimizing the Montgomery Modular Multiplier for a Power- and Area-Efficient Hardware Architecture	IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'20)	2020
8	2^0	Optimizing Iterative-based Dividers for an Efficient Natural Logarithm Operator Design	IEEE Latin American Symposium on Circuits and Systems (LASCAS'20)	2020
9	2^0	An Efficient N-bit 8-2 Adder Compressor with a Constant Internal Carry Propagation Delay	IEEE Latin American Symposium on Circuits and Systems (LASCAS'20)	2020
10	2^0	Energy-Efficient Haar Transform Architectures Using Efficient Addition Schemes	IEEE Latin American Symposium on Circuits and Systems (LASCAS'20)	2020
11	2^0	Exploring Architectural Solutions for an Energy-Efficient Kalman Filter Gain Realization	IEEE International Conference on Electronics, Circuits and Systems (ICECS'19)	2019
12	1^0	Maximizing the Power-Efficiency of the Approximate Pruned Modified Rounded DCT Exploiting Approximate Adder Compressors	IEEE International New Circuits and Systems Conference (NEWCAS'19)	2019
13	1^0	Exploring Motion Vector Cost with Partial Distortion Elimination in Sum of Absolute Differences for HEVC Integer Motion Estimation	IEEE International New Circuits and Systems Conference (NEWCAS'19)	2019

Table 6.4: Published International Conference Papers (from 2016 up to 2018).

#	O	Title	Conference	Year
14	2^0	Low-Power HEVC 8-Point 2-D Discrete Cosine Transform Hardware Using Adder Compressors	IEEE International New Circuits and Systems Conference (NEWCAS'18)	2018
15	2^0	Exploiting Partial Distortion Elimination in the Sumof Absolute Differences for Energy-Efficient HEVC Integer Motion Estimation	Symposium on Integrated Circuits and Systems Design (SBCCI'18)	2018
16	2^0	A Fixed-Point Natural Logarithm Approximation Hardware Design Using Taylor Series. In: New Generation of Circuits and Systems	New Generation Circuits and Systems Conference (NGCAS'18)	2018
17	2^0	Pruning and approximation of coefficients for power-efficient 2-D Discrete Tchebichef Transform	IEEE International New Circuits and Systems Conference (NEWCAS'17)	2017
18	2^0	A Power-Efficient 4-2 Adder Compressor Topology	IEEE International New Circuits and Systems Conference (NEWCAS'17)	2017
19	2^0	Low Power SATD Architecture Employing Multiple Sizes Hadamard Transforms and Adder Compressors	IEEE NEW Circuits and Systems Conference (NEWCAS'17)	2017
20	1^0	A power-predictive environment for fast and power-aware ASIC-based FIR filter design	Symposium on Integrated Circuits and Systems Design (SBCCI'17)	2017
21	2^0	Physical implementation of an ASIC-oriented SRAM-based viterbi decoder	IEEE International Conference on Electronics, Circuits and Systems (ICECS'17)	2017
22	1^0	Using adder and subtractor compressors to sum of absolute transformed differences architecture for low-power video encoding	IEEE International Conference on Electronics, Circuits and Systems (ICECS'17)	2017
23	2^0	Exploiting absolute arithmetic for power-efficient sum of absolute differences	IEEE International Conference on Electronics, Circuits and Systems (ICECS'17) (Submited)	2017
24	2^0	Improved Goldschmidt algorithm for fast and energy-efficient fixed-point divider	IEEE International Conference on Electronics, Circuits and Systems (ICECS'17),	2017
25	1^0	Using efficient adder compressors with a split-radix butterfly hardware architecture for low-power IoT smart sensors	IEEE International Conference on Electronics, Circuits and Systems (ICECS'17)	2017
26	2^0	Power-efficient sum of absolute differences architecture using adder compressors	IEEE International Conference on Electronics, Circuits and Systems (ICECS'16)	2016
27	1^0	A Power-Efficient Imprecise Radix-4 Multiplier applied to High Resolution Audio Processing	IEEE International Conference on Electronics, Circuits and Systems (ICECS'16)	2016