DESIGN AND OPTIMIZATION OF APPROXIMATE LOW POWER MULTIPLIER FOR DSP APPLICATION

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

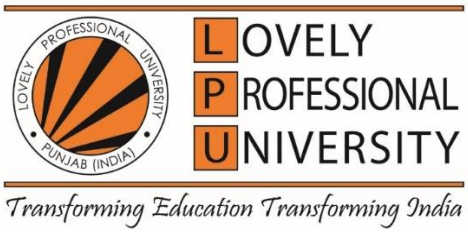
Electrical and Electronics Department

By

SHEETAL NAGAR

Registration Number: 41900673

Supervised By	Co-Supervised by	Co-Supervised by
Dr. Shanky Saxena	**Dr. Govind Singh Patel**	**Dr. Seema Nayak**
EE(Assistant Professor)	**EC(HOD)**	**ECE(HOD)**
LPU, Punjab	**SIOTCOE, YADRAV,MH**	**IIMTCOE, Greater Noida**



LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2025

DECLARATION

I, hereby declared that the presented work in the thesis entitled "**Design and optimization of Approximate low Power multiplier for DSP Applications**" in fulfillment of degree of **Doctor of Philosophy (PhD.)** is outcome of research work carried out by me under the supervision, Dr. Shanky Saxena working as Assistant Professor, in the Department of Electrical and Electronics of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgement have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

(Signature of Scholar)

Name of the scholar: Sheetal Nagar

Registration No.: 41900673

Department/school: Electronics and Electrical

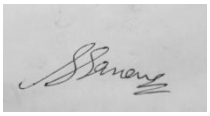
Lovely Professional University,

Punjab, India

Certificate

This is to certify that **Ms. Sheetal Nagar** has completed the PhD. Electronics and Communication Engineering titled, **"Design and optimization of Approximate low Power multiplier for DSP Applications"** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original investigation and study. No part of this thesis work has ever been submitted for any degree.

This thesis is fit for the submission for the partial fulfillment of the condition for the awardof degree of Ph.D. in Electronics and Communication Engineering.



Signature of Supervisor

Dr. Shanky Saxena

Associate Professor

School of Electrical Engineering

Lovely Professional University Punjab

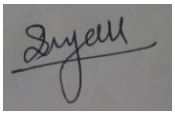
Signature of Co- Supervisor

Dr. Govind Singh Patel

Professor

Department Of ECE SITCOE ,

YADRAV,ICHALKARANJI, MH



Dr. Seema Nayak

Professor

Department of ECE,

IIMT College of Engineering Greater Noida

Abstract

In the modern era, the increasing demand for lightweight, low-power, and high-performance electronic devices has led to the development of advanced tools and techniques for designing energy-efficient multipliers. One such technique is compressor-based multiplication, which utilizes compression methods to reduce the number of partial products (PP) generated during the multiplication process, thereby decreasing the circuit's power consumption.This work focuses on the design of a compressor-based low-power multiplier, including its fundamental principles, implementation strategies, and potential applications. Furthermore, it explores the use of approximate computing techniques to further minimize power consumption in the compressor-based multiplier design. As a result, compressor-based low-power multiplication emerges as a promising approach for developing high-performance and energy-efficient multipliers, particularly suitable for battery-powered devices such as smartphones and light weight gadgets.

The proposed design incorporates a compressor-based Dadda multiplier to enhance key performance parameters. Both exact and inexact compressors are utilized to compress the partial product stages. The multiplier architecture integrates Half Adders (HA), Full Adders (FA), and both accurate and 4:2 inexact compressors. This architecture is implemented in an 8×8 Dadda multiplier to optimize power-related parameters. The work achieves notable reductions in delay and area, making these compressor-based designs especially suitable for error-tolerant or error-resilient applications.In today's fast-paced environment, which demands high-speed and compact solutions—particularly in wearable technologies—such multipliers are increasingly relevant. In this work, the design is also implemented for image processing applications. The multiplier is designed using Verilog, and the implementation tools include Cadence with 45 nm CMOS technology and Xilinx ISE. The proposed multiplier design demonstrates improved performance and efficiency, as visually summarized in the graphical abstract

A new era of the light weight, wearable which consume low power are on the top most demand of the society. Multiplication is the basic unit of the digital world. Consumption of low power in Multipliers is the highly popular in the research field. Inexact Computing is highly used methodology for the low power consumption in low power multipliers. This

Incredible fast technique most probable used the error tolerant applications like Video, Image Processing, FIR filters, Speech signal analysis and DSP (Digital signal processing) applications. The objective of this research is to explore the impact of inaccurate compressor designs on power utilization area in cell count and delay. The Proposed designs accoutered using 45 nm CMOS technology in Cadence. Through extensive experimentation and analysis, it was found that the proposed designs led to significant improvements in power consumption, with a reduction of 46%. Moreover, the delay was reduced by 65%, and there was a substantial decrease in area by 80.35% compared to accurate multiplier implementations. These findings underscore the potential of inaccurate compressor designs in enhancing the efficiency and performance of low-power multiplier Area and Power of proposed architecture had been obtained using the synthesis tool Xilinx Vivado using Verilog programming. Results of the work show the improvement in terms of Area and Power using cadence.

In the last overall conclusions from the present study has been presented along with the futurescope available with this study.

Acknowledgement

I extend my heartfelt gratitude to the individuals who have contributed, directly or indirectly, to the completion of this thesis.

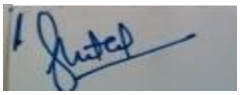
First and foremost, I am deeply thankful to my esteemed supervisors, Dr. Shanky Saxena, Dr. Govind Singh Patel, and Dr. Seema Nayak, for their invaluable advice, guidance, and unwavering support throughout my research journey. Their mentor-ship equipped me with the necessary skills for independent inquiry, fostering an environment where I could explore my ideas while receiving constructive feedback, ensuring the professional execution and timely completion of my work.

I am indebted to Dr. Shobit Saxena, Dr. Abhishek, Dr. Aruna Pathak, and Dr. P. Ramesh for their continuous technical assistance, which significantly enriched the quality of my research. Additionally, I express my gratitude to Dr. P. Ramesh for indirectly supporting my work through insightful research articles that served as constant sources of inspiration and guidance.

Attending various conferences and workshops during my PhD journey has been invaluable, and I am immensely grateful to all those with whom I collaborated and who provided invaluable feedback and constructive criticism.

Special thanks are due to Dr. Mayank Aggrwal, Dinesh Yadav, and Dr. Taslima Ahmad for their unwavering motivation and endless emotional support, which played a pivotal role in sustaining my enthusiasm and determination.

Finally, I extend my deepest appreciation to my friends and family for their unwavering love and support. My parents, Mr. Mahavir Singh Nagar and Mrs. Sheelam Nagar, have been pillars of strength, always encouraging me to strive for excellence. My husband, Mr. Jitendra Verma, and my daughter, Aashvi Verma, have been unwavering in their support, providing encouragement and understanding throughout this journey. Their unwavering support has been instrumental in my accomplishments thus far.The significance of emotional support in completing a PhD cannot be overstated, and I am truly grateful.



Sheetal Nagar

Table of Contents	

LIST OF FIGURE

LIST OF TABLE

ABBREVIATIONS

CMOS	Complimentary metal oxide Semiconductor
CPTL	Complimentary Pass Transistor Logic
CDAC	Capacitor-Based Digital-To-Analog Converter
CP	Charge Pump
DCO	Digitally Controlled Oscillator
DRC	Design Rule Check
DMRO	Dual-Mode Ring Oscillator
DPI	Digital Phase Interpolation
DPLL	Digital Phase Locked Loop
ECM	Error Correcting module
FA	Full Adder
GDI	Gate Diffusion Input
IC	Integrated Circuit
LE	Logic Element
LPM	Low Power Multiplier
NMOS	N-type metal oxide Semiconductor
PMOS	P-type metal oxide Semiconductor
PP	Partial Product
RC	Resistance Capacitance
VLSI	Very Large Scale Integration

CHAPTER 1
Introduction

1.1 MOTIVATION

Reducing power dissipation has become a top design priority due to the growing complexity of microelectronic circuits and the level of device integration [1]. Although power efficiency in electronic circuits has long been desired, it has only recently evolved into a crucial consideration during the initial stages of the design process [2]. Two types of power dissipation constraints exist. The first is related to cooling issues in high-performance systems [3]. High-speed circuits release large quantities of energy in a short period of time, generating significant heat as a byproduct. The integrated circuit package used must be capable of dissipating this heat. If the package (PCB, system enclosure, or heat sink) cannot sufficiently disperse the heat, or if the necessary thermal components are prohibitively expensive, heat removal may become a limiting constraint [4].

The second scenario of high-power circuit failure is linked to the rising acceptance of portable electronics. All battery-powered devices, including laptops, pagers, portable video players, and mobile phones, have a finite operating period before requiring recharging [5]. Low-power operation is preferred in integrated circuits to extend battery life. Additionally, newer generations of applications often demand more computational power, which places additional stress on the system's energy storage components [6]. Recent technological advancements have successfully reduced power usage. Power dissipation has decreased due to trends such as the shift from bipolar devices to CMOS and the shrinking of lithographic feature sizes, even though high integration speed was the primary driver of these advancements [7]. Power optimization has only recently been recognized as a separate design objective. In this study, we will focus on the digital multiplier, a component widely used in contemporary microelectronic circuit design, for power optimization [8]. In this chapter, we will provide a broad overview of power optimization and discuss how it relates to existing design processes. We will also give a quick overview of the

multiplier, its various uses, and some design objectives for creating efficient designs. With the rising integration of devices and the increasing complexity of electronic circuits, reducing power dissipation has become a primary design objective [9]. While power efficiency has always been advantageous in electronic circuits, it has recently turned into a limiting factor for a wide range of applications, requiring consideration from the outset of the design process [10]. Arithmetic circuits, such as multipliers and adders, are fundamental components in the design of communication circuits. Recently, there has been a growing interest in designing digital systems for communication applications and advanced signal processing with low power consumption and no performance penalty [11]. Designing low-power, high-speed arithmetic circuits requires a combination of techniques at four levels: algorithm, architecture, circuit, and system levels [12]. This study presents the design and simulation of a multiplier architecture suitable for high-performance and low-power applications.

Digital multipliers are the most commonly used components in various modern circuit designs. They are fast, reliable, and efficient components used to perform arithmetic operations [13]. Depending on the design of the components, there are different types of multipliers available. The choice of multiplier architecture is based on the specific application [14]. Power dissipation in a multiplier is a critical issue, as it reflects the total power consumed by the circuit and, thus, affects the performance of the device.

Current digital systems may require a high volume of calculations while facing certain energy and speed limitations [15]. In portable systems, where the energy stored in the battery is the primary source of power for operation, reducing energy consumption is a vital design objective. Similarly, this importance extends to systems designed to harness energy from the environment [16]. In numerous applications, such as communications, biomedical devices, multimedia, and the Internet of Things (IoT), the computations involve digital signal processing (DSP) of signals. For a large set of applications where minimal output quality is acceptable, DSP blocks may perform the necessary processing efficiently [17]. These blocks consist of arithmetic units that can operate using the approximate computational paradigm.

In these applications, the base output quality may depend on a trade-off between quality and energy efficiency/speed. Inexact computing inherently involves sacrificing precision for energy savings or speed (performance) [18]. This paradigm can be applied in both software and hardware domains of processing systems. In the hardware domain, several inexact components, such as adders and multipliers, have been introduced [19]. Notable examples of inexact adders include Estimated Arrival I, AMAs, TGAs, LOA, Estimated Arrival II, LREA, Stuff, iRAP-CLA, and QuAd. These inexact components have been evaluated using them in DSP blocks such as Finite Impulse Response (FIR) filters and Discrete Cosine Transform (DCT) applications, including image processing [20]. Inexact computing can also be applied at the algorithmic level in DSP applications, while precise components are used to execute data path tasks after algorithmic approximations have been applied [21]. The use of inexact units in computation frameworks, including DSP blocks, may degrade the output quality, which should be carefully assessed for optimal utilization of these units [22]. More specifically, one must evaluate the impact of estimation errors on output quality as a critical step in employing inexact units.

An efficient assessment can be accomplished using a mathematical model for the units, such as inexact adders. In traditional approaches, statistical representations of inexact component errors have been obtained through comprehensive simulations. To address the complexity, some researchers have relied on Monte Carlo (MC) simulations to determine quality [23]. However, the limitations of exhaustive and MC simulations include the need to simulate every type of multiplier along with its configuration, requiring substantial runtimes for evaluating the output quality of DSP applications [24]. Additionally, these methods do not provide insights into the sources and causes of the errors, nor do they offer a clear understanding of how estimation errors affect signal characteristics [25].

To overcome these issues, a fully analytical modeling approach for various error metrics, such as error probability and mean error distance, has recently been introduced [26]. Although the main problems were addressed in these studies, the use of convolution algorithms resulted in significant computational overhead in

evaluating output quality [27]. By employing MC simulations to find statistical error characteristics of the adders/multipliers, semi-analytical approaches have been introduced to address the extensive runtimes for output quality assessment. Ultimately, optimizing speed and energy characteristics for a given maximum acceptable output quality loss is a key objective in designing digital systems based on inexact computing [28]. Since the improvement structure heavily relies on the quality assessment technique, an efficient modeling procedure is highly desired to prevent excessive runtime [29]. Most digital signal processing (DSP) systems incorporate a multiplication unit to perform calculations such as correlation, convolution, filtering, and frequency analysis. In many DSP algorithms, the multiplier is in the critical delay path and ultimately determines the performance of the computation [30]. The speed of multiplication is crucial not only in DSP but also in general processors today, especially with the rise of media processing.

In the past, multiplication was primarily executed using a series of additions, subtraction, and shift operations. Recently, numerous multiplication algorithms have been developed, each with its own advantages and disadvantages in various fields [31]. The multiplier is a relatively large block within a computing system. The amount of hardware involved is proportional to the square of its size; for example, a multiplier of size n bits has $O(n^2)$ gates. For multiplication algorithms used in DSP applications, latency and throughput are the two significant constraints from a delay perspective [32]. Latency refers to the actual delay in processing a function, measuring how long it takes after the inputs to a device are stable for the results to be available on the outputs. Throughput is the number of multiplications that can be performed in a given period [33].

Multipliers are not only high-latency blocks but also significant sources of power dissipation. Therefore, if one aims to minimize power consumption, it is crucial to identify strategies to reduce delay through various optimization techniques [34]. Minimizing power consumption in digital systems involves optimizing all levels of design. This includes the technology used to implement the digital circuits, the style

of circuits and geometry, the architecture for executing the necessary circuits, and, at the highest level, the algorithms used [35].

The key technological concept revolves around supply voltage and voltage scaling, enabling the reduction of input voltage with minimal impact on logic speed [36]. Energy is consumed as capacitance is switched power can be reduced by minimizing capacitance through activity reduction, choice of number representation, careful management of signal transitions, re-synchronization to minimize glitches, logic design, circuit design, and physical layout [37].

High-level optimization strategies for low-power multipliers are covered in this dissertation. High-level approaches are algorithmic and architectural techniques that take into account the mathematical properties of multiplication and the characteristics of the input data [38]. This study's primary research hypothesis is that high-level multiplier design optimization results in more power-efficient solutions than low-level multiplier design optimization alone. We specifically explore how to regulate the active multiplier resources to align with the features of external data, as well as how to improve the internal algorithm and design of multipliers. Power reduction with minimal area and delay overhead is the main goal [39]. Another advantage of high-level optimization lies in its capacity to decrease both power consumption and area or delay through the use of novel algorithms or designs.

1.2 Power Optimization

Energy is a measure of all the Joules dissipated by a circuit, whereas power is the total number of Joules dissipated over a certain period of time. Although they are similar, low-power and low-energy designs have different objectives. Power becomes an issue primarily when cooling is a concern. Peak power is often used for reliability analysis, signal noise margin, and power and ground cable design [39]. In the context of improving battery lifespan, energy per operation or task is a better indicator of a system's energy efficiency. To evaluate the benefits of designs in digital CMOS, the well-known power-delay product is frequently utilized. However, this can be misleading since power × delay = (energy/delay) × delay = energy, implies that delay

is irrelevant. The term "energy-delay product" should be used instead, as it refers to two separate measurements of circuit behavior. To ensure that a comparison of power-delay products is equivalent to a comparison of energy-delay products, various schemes should be evaluated at the same frequency.

Dynamic power and static power are the two main sources of power dissipation in digital CMOS circuitry. Dynamic power is associated with capacitance charging and discharging, short-circuit current dissipation, and other events that change logic states. The sole source of static current in CMOS logic is unintentional leakage current, which may manifest as reverse-biased PN junction current or sub-threshold channel conduction current [34]. Even so, sporadic exceptions to the conventional CMOS logic, including pseudo-N MOS logic, can result in significant static current.

Power optimization of digital systems has been researched at various abstraction levels, from the most fundamental technological level to the most complex system level [35]. Power consumption is reduced at the technological level through advancements in the manufacturing process, including smaller feature sizes, extremely low voltages, copper interconnects, and insulators with low dielectric constants [36]. Lower voltages can be applied to non-critical system blocks thanks to the support for different supply voltages during manufacturing [37]. To lessen imbalances in wire capacitance and signal latency, placement and routing are modified at the layout level. Transistor sizing, network restructuring, and other circuit logic designs are employed to reduce power at the circuit level.Various strategies have been proposed at the gate level. Gate-level techniques used in commercial tools like Power Compiler include cell size, cell composition, equivalent pin switching, and buffer insertion. These approaches yield an average power saving of 11% while increasing area by 9%. Additional gate-level techniques, such as signal gating, delay balancing, input synchronization, and signal polarity optimization, have also been suggested.Clock gating has been widely used at the register-transfer level (RTL) to deactivate combi-national or sequential blocks that are not in use at a given time. In Power Compiler, operand isolation is employed to deactivate a data path unit if its outputs are not being used, while clock gating disables an entire register bank.

Computational gating is an RTL signal gating method that identifies output-invariant logical conditions at some combi-national block inputs and inhibits inputs under these conditions. Re timing rearranges registers in sequential circuits to prevent the propagation of erroneous transitions [38].

There is considerable flexibility in power optimization at the architectural and system levels. The two primary methods for achieving better performance than required and then trading operation frequency for supply voltage reduction are parallelism and pipe-lining. Transforming the control-data-flow graph (CDFG) is another effective strategy for creating low-power architecture [39]. Asynchronous systems are also studied to avoid a global clock signal and minimize unnecessary computations. The practice of deactivating or disabling components in event-driven systems when they are not in use is referred to as a power management approach.Despite the fact that power reduction has been achieved through optimization approaches at all levels, strategies at the lowest technology level and the highest architecture/system level are often more effective than those at intermediate levels. Three significant power consumption parameters—C_L and V_{DD}—are impacted by technology-level optimization. All four factors (C_L, V_{DD}, switching activity, and frequency) are influenced by algorithm/architecture-level optimization, particularly through the identification of candidates for V_{DD} reduction. In contrast, middle-level optimization often has a limited impact on one or two aspects.

1.3 Architecture of Low Power Multiplier Design

Multiplication consists of three phases: producing partial products (PPs), reducing partial products (PPR), and performing carry-propagate addition (CPA) [38]. In general implementations, there are sequential and combi-national multipliers. Due to the high level of integration now possible, this study focuses solely on combi-national multipliers used in digital VLSI systems. Several multiplication algorithms exist within the PPG, PPR, and CPA methodologies.

Radix-2 digit-vector multiplication is the simplest form for PPG, utilizing AND gates to reduce the number of PPs and thus the overall complexity. One operand is often encoded into high-radix digit sets based on area and delay considerations of PP reduction. The radix-4 digit set, with values {–2, –1, 0, 1, 2}, is the most commonly used. For PPR, there are two approaches: column reduction, carried out by an array of counters, and row reduction, performed by an array of adders. There are two main categories within row reduction: linear arrays and tree arrays. While both tree array and column reduction have delays of O(log n), where *n* is the number of PPs, linear arrays have a latency of O(n). A fast adder technique is necessary for the final CPA since it is on the critical path. If it is desirable to retain duplicate results from PPG for additional arithmetic operations, the final CPA may occasionally be delayed.

Low-level strategies, particularly many of the power optimization techniques discussed in Section 1.2, are relevant to the design of low-power multipliers [35]. These low-level methods have been examined for multipliers and include signal polarity optimization, delay balancing, input synchronization, transistor reordering and sizing, pass-transistor logic, swing-restricted logic, transistor sizing, and voltage scaling [36]. However, these methods have only moderately improved power consumption in multipliers, often requiring significant design effort or resulting in high area or delay overhead.

Three factors make low-power multiplier design challenging. First, there is a quadratic relationship between operand accuracy and multiplier area. Second, the multiple logic levels in parallel multipliers can cause erroneous transitions or errors [37]. Third, to achieve high speed, parallel multipliers may have highly complex structures, which reduces the effectiveness of layout and circuit-level optimization.

Multiplication is a fundamental arithmetic operation that differs from random logic in several ways at both the algorithmic and bit levels of computing. However, low-level power optimization techniques have not adequately addressed these differences [38]. Additionally, the low-level properties of the incoming data are challenging to account for. Therefore, it would be beneficial to develop algorithmic and architectural-level

power optimization strategies that consider the mathematical properties and operand characteristics of multiplication [39].

Multipliers perform the core function of multiplication in most arithmetic computing systems [40]. They are an indispensable part of various applications such as FIR filters, arithmetic circuits, the Internet of Things (IoT), DSP applications, artificial intelligence, and other domains that involve intensive computation. However, multipliers consume a significant amount of power, which in turn reduces processing speed. Reducing power consumption in multipliers can help shorten processing time and improve overall system efficiency [42]. As a result, low-power design solutions for multipliers have generated significant interest in recent years. Based on the techniques used to perform multiplication, multipliers can be categorized as follows:

1. ARREY Multipliers
2. BOOTH Multipliers
3. WALLACE Multipliers

These multipliers are suitable for Very Large Scale Integration (VLSI) technology with CMOS implementation. Array multipliers are fast, but they suffer from high hardware complexity. The area required for array multipliers is large, which leads to increased delay and, consequently, higher power consumption. Booth multipliers are used for high-speed applications; however, their main drawback is excessive power consumption. Wallace multipliers are also designed for fast operations and are commonly employed in speed-critical systems. Some ancient Vedic methods of multiplication include:

1. Urdhva Tiryakbhyam sutra
2. Nikhilam sutra

These Vedic multipliers are particularly effective when handling a large number of bits. The Urdhva Tiryakbhyam Sutra is based on traditional multiplication techniques [43]. Both Vedic methods are known for their high processing speed.Vedic multipliers are fast and widely applicable in DSP applications. As the number of bits

(N) increases, delay is significantly reduced in Vedic multipliers, making them well-suited for modern digital designs. The main aim of this paper is to present a comprehensive literature review of different multiplier designs and their applications.

1.3.1 Architecture of multipliers

The dynamic growth of DSP applications is increasing day by day due to the high demand for low power consumption, less area, and lightweight designs, which are top priorities for designers working on communication systems and multimedia devices [44]. To design low-power multipliers, various tools are employed for simulation, synthesis, and parameter analysis. Multipliers can be implemented using tools like Xilinx 14.5, Xilinx ISE 14.7, Cadence, and Verilog HDL, depending on the target application area.

The Wallace Multiplier uses a column compression architecture to improve multiplication speed. Its performance is significantly better than that of the Array Multiplier, particularly because the delay variation depends on the length of the operand word. XOR- X NOR based compressors like 5:2, 4:2, and 3:2 are used to reduce the number of partial products in the Wallace Tree structure [45].

In the case of the Dadda Multiplier, as the operand word size increases, so do the key performance parameters—speed, area, and power consumption [46]. Traditional 8×8 Dadda Multipliers require extensive wiring overhead due to a large number of addition operations. The authors compared several multiplier architectures, including Regular Dadda, Decomposed Dadda, Partitioned Dadda, Wallace Tree (based on 3:2, 4:2, and 5:2 compressors), and Hybrid Multiplier designs. The simulation results of these architectures yielded a total power consumption ranging from 0.737 to 2.440 mW, delay from 4.4 to 7.5 ns, and a power-delay product between 5.527 and 10.73 [47]. The performance of digital multipliers such as Carry Save, Array, Baugh-Wooley, and Wallace Tree Multipliers was also analyzed. A 0.8V power supply and 22nm Predictive Technology Models were used for simulation. According to the analysis, the Baugh-Wooley Multiplier had the lowest delay and emerged as the fastest multiplier. In this method, the authors implemented a 2’s complement

approach for multiplication and presented results for a 4×4 Baugh Wooley Multiplier. Using a 46-transistor full adder cell, the Array Multiplier had a delay of 165 ps and power consumption of 135 μW, while the Baugh Wooley Multiplier had a delay of 123.26 ps and power of 85.11 μW. When implemented with a 28-transistor full adder, the Array Multiplier showed 154.68 ps delay and 88.42 μW power, while the Baugh-Wooley Multiplier showed 106 ps delay and 79.63 μW power [48]. The analysis also considered the Wallace Tree and Carry Save Multipliers.

A high-speed and area-efficient Booth Re-code Wallace Tree Multiplier was proposed for fast arithmetic circuits. In this design, the author reduced the area of the Wallace Tree Multiplier by applying the Booth Re-coding Algorithm and a compressor adder. The design was implemented using Verilog HDL, and synthesis was performed on a Xilinx Virtex-6 FPGA. Results showed that the proposed Wallace Tree Multiplier is 67% more efficient in terms of speed compared to the existing Wallace Tree architecture. Additionally, it was 53% faster than the Vedic Multiplier, 22% faster than the Radix-8 Booth Multiplier, and 18% faster than the Radix-16 Booth Multiplier. The area of the proposed Wallace Tree Multiplier was also significantly reduced [49].

Multiplication is a fundamental operation in all microprocessor and micro controller-based systems used in DSP applications and is typically performed using combi-national circuits [20]. Based on the literature review presented in this paper, the focus remains on the low-power, high-speed, and area-efficient design of multipliers. Among the various architectures proposed, the Wallace Tree and Dadda Multipliers are the most frequently considered due to their strong performance across critical parameters. Consequently, this study focuses on the architectures and analysis of these multipliers.

1.3.2 Wallace Tree Multiplier: Wallace Tree Multipliers were developed by Wallace to enable the fast processing of two-number multiplication [49]. In the Wallace Tree Multiplier method, a three-step multiplication process is applied. A three-input Wallace Tree circuit uses Carry-Save Adders (CSA), as shown in Figure 1.1.The output from each stage is supplied as input to the next stage, allowing for efficient reduction of partial products and accelerating the overall multiplication process.

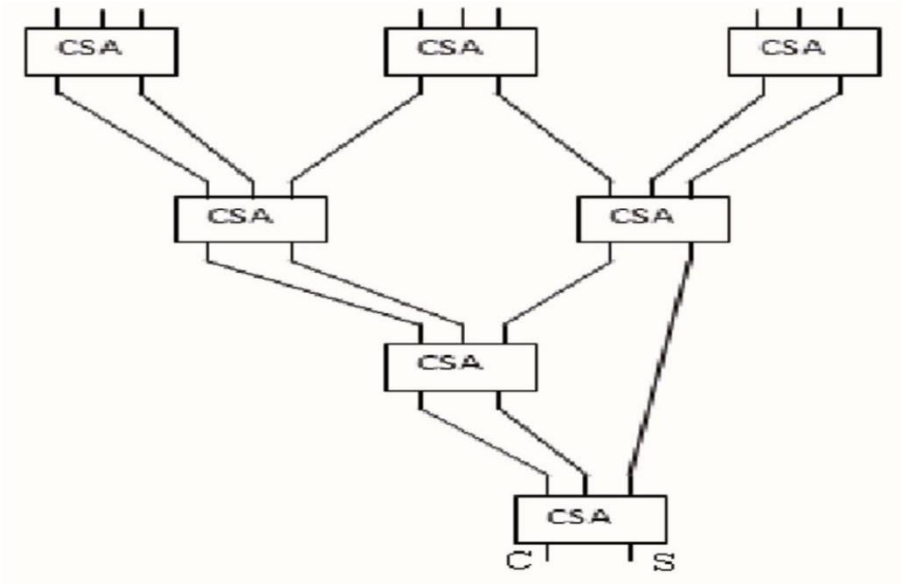


Figure 1.1 Wallace Tree Multiplier [49]

The circuit layout in the Wallace Tree Multiplier method is somewhat complex, but it offers very high speed performance [12]. Its fast switching speed and low power consumption make the Wallace Tree Multiplier a high-speed multiplier with a superior architecture [49]. There are different types of Wallace Tree Multipliers, each designed to further optimize specific performance parameters such as area, delay, or power efficiency.

1.3.3. Conventional Wallace tree Multiplier: This is the simplest multiplier in terms of architecture, consisting of three main blocks. The first block handles the generation of partial products, the second block is responsible for the accumulation of partial products, and the final stage performs the addition, as illustrated in Figure 1.2.

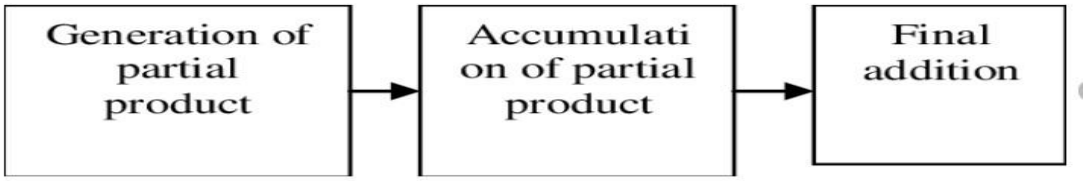


Figure 1.2.Conventional Wallace Tree Multiplier [49].

1.3.4. Modified Wallace tree Multiplier: In this Wallace tree multiplier Reverse pyramid style is applied by the designers for partial products. A reduction tree to reduce the complexity of Wallace tree multiplier [50]. Figure 1.3 shows the modified Wallace tree multiplier is using in a MAC unit.

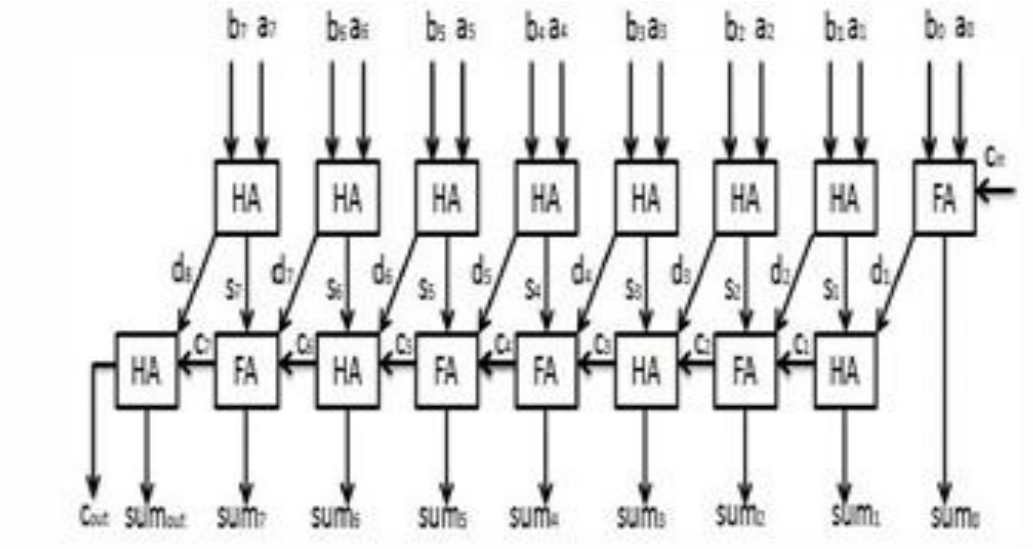


Figure 1.3. Modified Wallace Tree Multiplier using in MAC Unit [50].

1.3.5 Wallace Booth Multiplier: In this multiplier, the Booth Recoding algorithm and compressors are used. The Booth Recoding algorithm is employed to reduce the number of partial products [29]. The main advantage of this architecture is its reduced area and high speed [35].A Wallace Tree architecture is implemented using compressors instead of traditional full adders. The architecture of the Wallace Tree multiplier is modified using 3:2 compressors in the first stage, 4:2 compressors in the second stage, and 5:2 compressors in the third stage, essentially forming a combination of three full adders.Figure 1.4 illustrates the Novel Wallace Booth Multiplier.

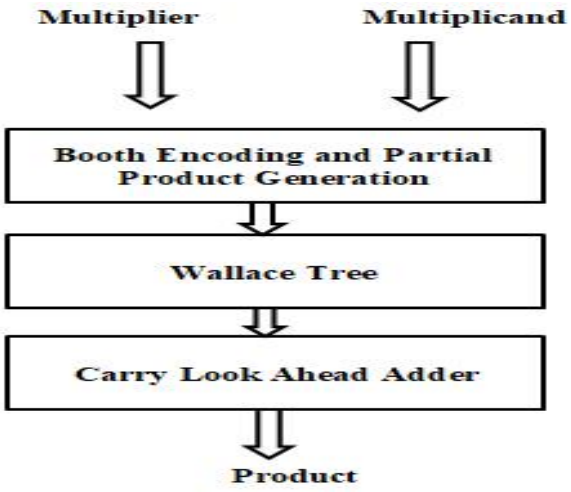


Figure 1.4. Novel Wallace Booth Multiplier [51].

1.3.6 Dadda Multiplier: The Dadda Multiplier was invented by computer scientist Luigi Dadda in 1965 [51]. It is derived from the family of parallel multipliers. In Dadda multipliers, the adder stages are minimized, making it an advanced and refined version of the Wallace multiplier.The Dadda multiplier is implemented based on the Dadda scheme, which reduces the number of adder stages required for the addition of partial products. Full adders and half adders are used in this architecture. The structure of the Dadda multiplier reduces the tree to only two rows, allowing for stage-wise reduction as necessary. Overall, the Dadda multiplier is a refinement of the Wallace multiplier, optimized for area and delay. A 4×4 Dadda multiplier using half adders and full adders along with a Ripple Carry Adder (RCA) at different stages is shown in Figure 1.5.

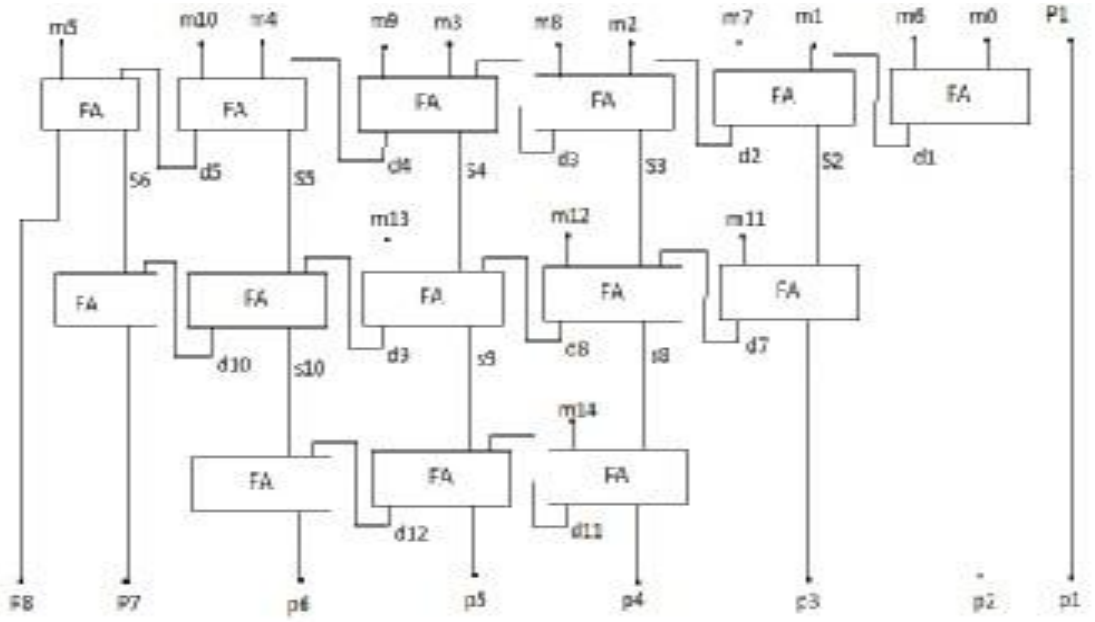


Figure 1.5. 4x4 Dadda Multiplier using RCA [51]

Requirement of wider and fast CPA is the main drawback of Dadda Multiplier. Structure of Wallace tree multiplier is better than the Dadda multiplier [51].

The Hybrid multiplier is an advanced research topic in multiplier design, combining both Wallace multiplier and Dadda multiplier architectures. A 4×4 Dadda multiplier and a 4×4 Wallace multiplier were constructed for analysis and simulation. The simulation tool used was MICROWIND with a technology node of 0.25 micrometers. Various parameters of these multipliers were simulated, and notably, compressors were not used in this structure. The design consumes less area and low power, showing a 40% improvement in the power-delay product compared to the existing 8×8 Wallace and Dadda multipliers, which use compressors.

This dissertation systematically examines low-power multiplier design at both the algorithmic and architectural levels from two perspectives: internal efforts focused on multiplier design and external efforts considering input data characteristics. For internal efforts, we consider techniques such as PPG re-coding, operand representation, and PPR structure optimization. For external efforts, we investigate signal gating to disable portions of a full-precision multiplier. The types of multipliers researched include linear array multipliers, leapfrog multipliers, left-to-right multipliers, split multipliers, and tree multipliers. Column reduction multipliers are not considered, as they achieve similar latency and area as tree arrays but significantly worsen regularity. Since adding signed integers is convenient using the two's complement form, it is widely employed in digital arithmetic systems.

1.4 Integrated Circuit Multiplication

Digital multiplication is one of the most fundamental operations in a wide variety of algorithms. The widespread use of the multiplication operation in computing has led to numerous implementations with various requirements and objectives. Some applications require a wide dynamic range and high accuracy, while in other cases, these requirements are not as strictly defined [52]. When high accuracy is a concern, digital multiplication is preferred over analog methods, as it is relatively straightforward to create digital multipliers that meet the specific accuracy required by the application.

The level of accuracy necessary for multiplication varies by function. For example, image compression methods typically require 8 bits, whereas more accurate digital signal processing (DSP) tasks require 16 bits. Generally, digital signal processing utilizes 16-bit multipliers, while microprocessors employ 53/64-bit multipliers. In these designs, integer multiplication is the fundamental operation; in floating-point multipliers, the integer multiplication units are smaller versions of the larger floating-point units [53]. The architecture is influenced by whether the procedures are signed or unsigned, and several innovative methods have been proposed to modify the bit representation of numbers to achieve power savings. However, delay has historically been, and remains, the primary concern in multiplication.

1.5 Design Consideration of Digital Circuits

In ensuring correct digital functionality, the primary concern for system designers has always been speed. A circuit is designed to operate at a specific delay; if this delay is not met, the entire system may fail to function properly. While further reductions in delay are beneficial, they are not always strictly necessary.

Other factors may hold equal or greater importance than power dissipation; considerations regarding area, performance, and reliability are also crucial for designers. It’s worth noting that power reduction methods are not necessarily negatively correlated with delay reduction. For instance, one strategy to reduce delay in a circuit is to increase the driving strength of gates, which can lead to increased power consumption [53]. However, reducing interconnect capacitance is another method that lowers both power and delay. Generally, significant power savings can be achieved if delay is not a critical issue, but it is irrelevant to prioritize power reduction without considering other factors.

1.6 Power vs Energy

For the purposes of this discussion, the distinction between the terms "power" and "energy" is crucial. It should be noted that power refers to the rate at which energy is dissipated by a circuit over time, whereas energy measures the total amount of energy

dissipated by a circuit regardless of the time factor. Power reduction is technically a distinct objective from energy reduction. When heat removal is an issue, power can become problematic.

A package's heat sink might not be able to transfer heat quickly enough if too much energy is converted into heat in a short period; this can result in an increase in temperature and subsequent thermal failure. The cooling structure is better equipped to handle the circuit's thermal demands if the same amount of energy is generated over a longer period of time [53]. Here, power reduction refers to minimizing the situation in which a significant amount of energy is lost in a short time frame. However, the overall amount of energy lost may not change.

The goal of energy reduction is to lower the total amount of energy that must be dissipated. Since time is not a factor in this calculation, we often refer to energy per operation as the metric to be optimized. Thus, while energy reduction pertains to optimizing battery longevity, power reduction falls within the realm of thermal reliability.

1.7 Delay and Power in Multipliers

The multiplication operations used by digital signal processing (DSP) algorithms are the main topic of this thesis. In this domain, accuracy is often required to be 8 or 16 bits. Algorithms impose two limitations on multiplication in terms of delays: throughput and latency. A device's latency measures how long it takes for the final result to appear on the outputs once the inputs are stable. Throughput, on the other hand, measures how many multiplications can be completed in a specific length of time . For a straightforward combi-national multiplier, throughput depends on delay. However, there are several methods for computing multiple multiplications simultaneously, where latency is only tangentially connected to throughput.

Throughput is a critical issue in DSP. Digital signal processing algorithms frequently deal with the transmission and detection of audio and visual inputs, such as images or sounds [54]. In such applications, using narrow bit-width multipliers is feasible

because the precision requirements are less stringent compared to other tasks, such as numerical algorithms for scientific computing. Additionally, small multipliers do not lead to lengthy delay paths. The key limiting specification for many DSP applications is throughput . These tasks often require processing images of relatively low quality, but they work with vast amounts of data representing various image or sound samples. For instance, generating an image involves executing calculations on numerous polygons, but the required accuracy (in bits needed to identify a specific color or spatial coordinates) is minimal. Numerous 8-bit computations are necessary for voice compression [55].

One technique to execute a large number of calculations quickly is to reduce the latency of multiplication, allowing the multiplier to start executing the next operation sooner. However, increasing throughput is a more effective approach to boost the number of calculations. Using multiple devices on a single chip increases throughput by the number of additional units, whereas pipeline or interleaving data allows a single functional unit to perform multiple tasks concurrently. These methods are typically more effective than merely reducing latency since attempts to decrease a circuit's delay often lead to diminishing returns (after a certain point, increasing the number of transistors does not significantly reduce delay). Conversely, optimizing throughput allows the number of operations that can be calculated in a given period to rise with each new functional block introduced.

In strongly coupled hardware/software systems, implementing pipeline can be challenging, even though it improves throughput. While the logic implementation of pipeline is relatively simple, it can be difficult to persuade compilers to create programs that effectively utilize it. The challenge lies in establishing a sequence of tasks to start execution while other tasks are still finishing. These "concurrent" or "overlapping" methods of system behavior introduce timing conditions that complicate the task of writing a compiler that can exploit such hardware techniques. Furthermore, while some DSP calculations lend themselves to parallel execution, others require processing to be more sequential, rendering additional pipeline hardware unnecessary [56]. Ultimately, very high-speed code is often written by hand

in low-level programming. Understanding all the techniques for optimizing a pipe lined function can be quite tedious when done manually.

These considerations support the use of multiple multiplication functional units on a single chip rather than implementing a heavily pipe lined multiplier. The multiplier is a relatively large block within a computing system. The amount of hardware involved is proportional to the square of its size (i.e., a multiplier of size nm bits has O(n2)O(n^2)O(n2) gates). Not only is the multiplier a high-delay block, but it can also be a significant source of power dissipation. Given the arguments outlined above, having several multipliers on-chip becomes essential as more DSP processing power is required [57]. The power dissipation associated with multiplication will become more pronounced, regardless of whether the pipe lined approach is used; a first-order pipe lined multiplier will dissipate as much power as several multiplier blocks. Although a pipe lined version has fewer gates, it will still experience a similar amount of switching. Consequently, digital multipliers have become one of the primary circuits targeted for power reduction.

1.8 Overview of Monograph

Following this, a concise overview of the research methodology employed in this dissertation is provided. The design and experimental procedures used in our research are fully described in the thesis. We start by identifying crucial elements that impact algorithmic and architectural power usage. These elements include the internal designs of multipliers and the characteristics of external data. Second, we propose advanced power optimization approaches that integrate both internal and external efforts. Internal work includes improvements to the reduction structure, operand representation, and multiplier re-coding, while external actions involve various forms of signal gating. To assess the feasibility of each method, we conduct high-level first-order estimates and theoretical analyses.

Third, we implement the proposed schemes and associated earlier schemes in structural verilog language that are independent of technology. For design optimization and maintenance flexibility this language is used widely. These designs

can be translated into the Artisan TSMC 0.18-mm, 1.8-volt standard-cell library and assessed using wire-load models in the Synopsys design environment or Cadence. The next step involves running automatic layouts with Cadence Silicon Ensemble, using guided floor-planning to examine the impact of layout on high-performance multipliers.

In this work, we aim to identify the relevant factors that affect both power and delay in multipliers. As referenced above, power should be considered simultaneously with delay; a reduction in power with a corresponding increase in delay yields net advantage. In the first phase of our research, we focused on existing delay reduction strategies for multipliers and examined their power dissipation properties. Delay reduction techniques for digital multipliers are well established, with a history spanning over thirty years. Various delay targets can be met using different enhancement strategies, and at times, multiple methods may be applied simultaneously. It is of great interest to identify which strategies should be employed to reduce delay while also aiming to minimize power consumption. Those approaches that show the most promise, as well as those that have not been previously explored, are typically the focus of our most intense investigation.

In the second phase of this work, we sought to expand the range of potential strategies by exploring ideas for power reduction that emerged from the initial analysis. Not all the ideas we implemented were successful; however, we identified several methods that show promise for reducing power in multipliers. Our experimental methodology relied on the standard cell design approach. These were thoroughly characterized for energy dissipation and delay under various input conditions and output loads. The multiplier logic was derived from standard descriptions found in the literature, and the actual logic synthesis was performed automatically using synthesis algorithms obtained from the literature, along with several enhancements we added to the process.

Once the multiplier was synthesized, the impact of physical effects was evaluated using a placement and routing stage, which aimed to determine rough wiring characteristics based on routing length. In later stages, results were validated by performing a full design using the Cadence tool suite. Given the logic and physical

description of the functionality, timing analysis was conducted using static delay data from the specified standard cells. Delay was estimated using a logic simulator that counted switching events and evaluated glitches.

Our design methodology evolved over time as our tools were refined and accuracy improved. First, we needed to perform specific tasks that were impractical using vendor tools—some of these, like multiplier synthesis, were essential to this dissertation's contributions. Second, while vendor tools could perform some tasks necessary for this work (e.g., Cadence could be used for static timing analysis), we required slight modifications to vendor tool functionalities (e.g., timing analysis computing the static longest path) that were difficult to achieve. Although coding such methodologies increased the time required to complete this task, we also gained the advantage of accessing these techniques at a very fine level of detail (for instance, we could integrate timing code within the inner loop of our multiplier synthesis algorithm).

The primary objective of this research is to explore and develop an efficient design framework for Approximate Low Power Multipliers, particularly suited for Digital Signal Processing (DSP) applications. To begin with, the study aims to establish a comprehensive set of operating parameters that will serve as a foundation for the classification and evaluation of various approximate multiplier designs. This classification facilitates a systematic approach to analyze trade-offs between accuracy, power, delay, and area. Following this, the research focuses on the development of a novel algorithm for low power multiplication that incorporates approximation techniques to achieve substantial energy savings while maintaining acceptable levels of computational accuracy. A key goal of the proposed work is to optimize power consumption in the designed multiplier architecture to enhance overall system performance, particularly in power-constrained environments. This architectural design will consider the critical balance between performance, power, and precision, ensuring practical applicability in real-world signal processing scenarios.

For more accurate delay and power computation, interconnect parameters are retrieved and back-annotated into Cadence. Full-timing gate-level simulation yields a

power estimation. Finally, experimental findings comparing various compressor based low power digital multiplier are presented and discussed.

1.9 Organization of the Thesis

The organization of thesis is as follows

Chapter 1 presents the introduction to the work, beginning with the motivation behind the research followed by the need for power optimization in digital circuits. The architecture of low power multipliers is explored in detail, including various multiplier structures such as the Wallace tree, Dadda, and Booth multipliers. Integrated circuit multiplication is discussed, along with important digital design considerations. A comparative overview of power versus energy, and delay and power trade-offs in multipliers is also presented. The chapter concludes with an overview of the entire monograph.

Chapter 2 provides a detailed literature review covering fundamental concepts of multipliers and their role in digital design. The structure and operation of multipliers are explained including partial product generation and reduction techniques such as array and Wallace tree reduction. The chapter further explores CMOS power dissipation, differentiating between static and dynamic power. The fundamentals of power optimization are addressed, followed by a review of existing literature to identify research gaps relevant to the thesis.

Chapter 3 outlines the optimization techniques and proposed methodology used in this research. It starts by describing the operating parameters used for categorization. A detailed explanation of the proposed algorithm, particularly the design and function of the accurate and approximate 4:2 compressor, is given. The compressor's integration into the Dadda multiplier is discussed. Power optimization strategies and an error-correcting module (ECM) are introduced. Circuit design considerations and performance analysis methods are also covered.

Chapter 4 presents the proposed design and model of the system. The design architecture and the tools used for implementation are explained. The proposed

system integrates an approximate compressor-based multiplier along with an ECM. The detailed design of the proposed Dadda multiplier is elaborated. Simulation results are presented and discussed to evaluate design efficiency and parameter optimization.

Chapter 5 covers the experimental results and performance evaluation of the proposed architecture. Performance analysis is carried out using Xilinx and Cadence tools under different library conditions (fast.lib and slow.lib). Parameter results are reported and analyzed. The chapter also examines the impact of optimization techniques on overall performance, including their effectiveness. A practical application in image processing is briefly introduced to demonstrate the proposed design's utility. Finally, the extent to which the research objectives have been achieved is discussed.

Chapter 6 concludes the thesis by summarizing the contributions of the research. It also outlines the future scope and potential extensions of the work in related areas.

The thesis concludes with references and a list of publications that support and relate to the work presented.

CHAPTER 2
Literature Review: Multipliers and Power Minimization Using Compressors

This Chapter presents about a portion of the essential ideas which are normal to the areas which we examined. To start with, a short depiction of computerized multipliers, their design and applications. Postpone decrease methods are likewise talked about. Then, we go over power scattering in CMOS circuits, alongside a few essential strategies which can be used to decrease power parameters.

2.1 Introduction to Multipliers

To comprehend postponement and parameters as power compromises in multipliers circuits, it will portray the essential circuit design of computerized multiplier executions . Inspect a portion of strategies it is created to diminish multiplier timings, especially acquire a comprehension of trademark power dispersal. Few understanding acquired using perception of rationale phenomenon, power dissemination reviewed by few sources; methods which decrease the parameter power because any of these sources can demolish the parameter power scattering by other one. Short conversation of wellsprings of power scattering in computerized CMOS delineate the significant impacts. Integration of total applicable power disseminating impacts this investigation, It is decided to assess multiplier plans by fostering a strategy which uses different methodology for circuits at low level test systems to compute power utilization. Make up to gradualness definite methodology, a reenactment based assessment program permitted fast examination of force and postponement to plans, exactness confirmed utilizing the point by point circuit test systems.

2.1.1 Basic structure of Multiplier

Essentially, automated replication should be evident to facilitate the advancement of spot movements that involve additions [58]. Here, two numbers, the multiplicand and the multiplier, are combined prior to the final addition.

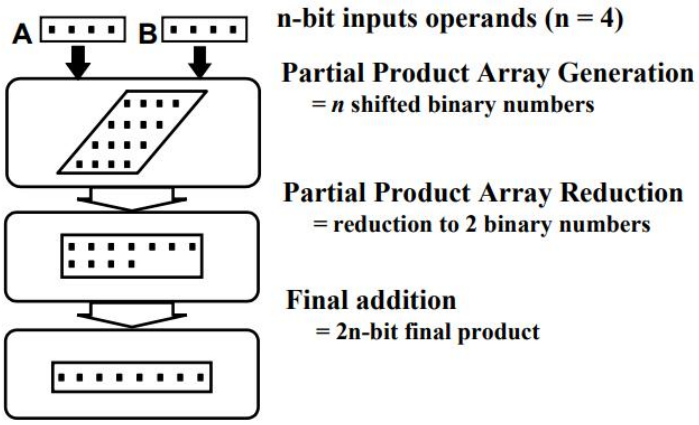


Figure 2.1. Digital multiplier flow [58]

Think about duplication of same numbers: the multiplicand C and multiplier P, now P is a n-cycle number using bit portrayal {pn-1, pn-2,...,p0}, main piece pn-1, most in-huge piece p0; multiplicand has a comparative piece portrayal {cn-1,cn2,...,c0}. Unsigned augmentation, for the n moved multiplicand are added with structure outcome. Whole strategy separated using stages: fractional item (PP) age, incomplete item decrease, and last expansion. This is delineated thoughtfully as shown in Figure 2.1.

2.1.2 Generation of PP(Partial Product)

Underlying move toward computerized duplication is to produce n moved duplicates of the C, it included the following steps [58]. A available moved duplicate of the C will added relies upon worth of P bit that is multiplier bit relating to C duplicate. In event that the multiplier and multiplicand are included.

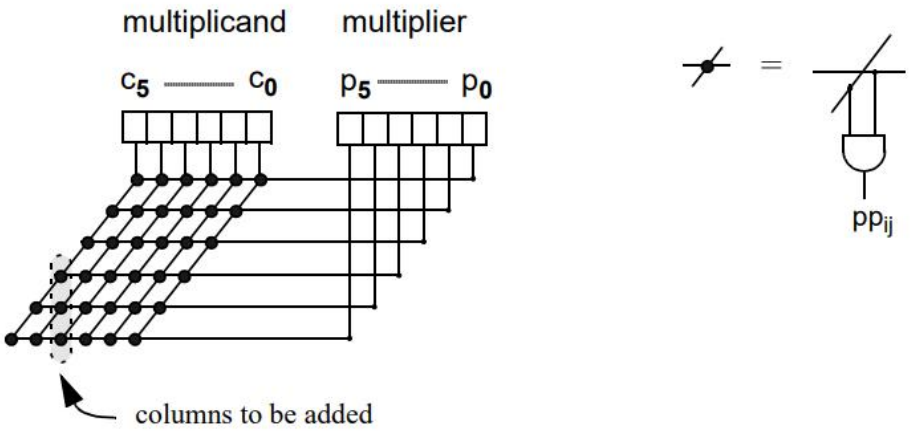


Figure 2.2.PP(Partial product) generation for 6*6 bit multiplication [58].

In the event that the piece is '0', C isn't add. Piece portrayal by capability will carried out utilizing coherent AND entryway, it performs AND(Pi,Ci) , j = 0 to n-1 , I = 0 to n-1. Subsequent qualities are called incomplete item bits or just, incomplete items; assuming we cover these halfway items by bit positions, we show up at a construction displayed in Figure 2.2. In this plan, the halfway item pieces are organized in segments. Subsequent trapezoidal construction is known as a fractional item cluster or basically PPA. Different types of incomplete item exhibits exist, contingent upon the number portrayal. For model, in the accompanying segment we depict a typical variation called Corner re coding, which permits a marked multi-bit incomplete item portrayal of the plan. Normal variations for proficiently carrying out two's supplement are depicted in [57]. There are a few significant focuses to see about the halfway item exhibit. To start with, in the most fundamental detailing (PPA bits produced through consistent AND), every one of the pieces are made in equal; Static postponement of every one pieces is equivalent. Next is components of the exhibit is elements of size of P and C : the level of the cluster is relative to the size of the P, and width are corresponding to size of C. At long last, every one of the pieces in a specific section are added, & a few segments have less pieces than other ones.

2.1.3 Reduction of Partial Product

Basics of a productive advanced multiplier execution is how PPA pieces added. Traditional convey adders carry out addition tasks, deferral of multitude of adders consumes more time. All things being equal, the incomplete item is decreased utilizing a procedure called convey save expansion [58]. This methodology permits progressive increments to be integrated into one worldwide expansion step. Think about a mathematical piece vector portrayal of the accompanying structure.

Think about the case where good for nothing vectors added . The most reduced piece state, 2 pieces are add then convey is proliferated to following piece state. That point convey in & following 2 pieces at maximum piece position consolidated, & do produced. Utilizing undulating strategy, addition of two n-digit number taken O(n) successive piece options, coming about in postponement O(n).We need addition of three piece a,b and c, every one size is n, then utilize it to addition of initial.

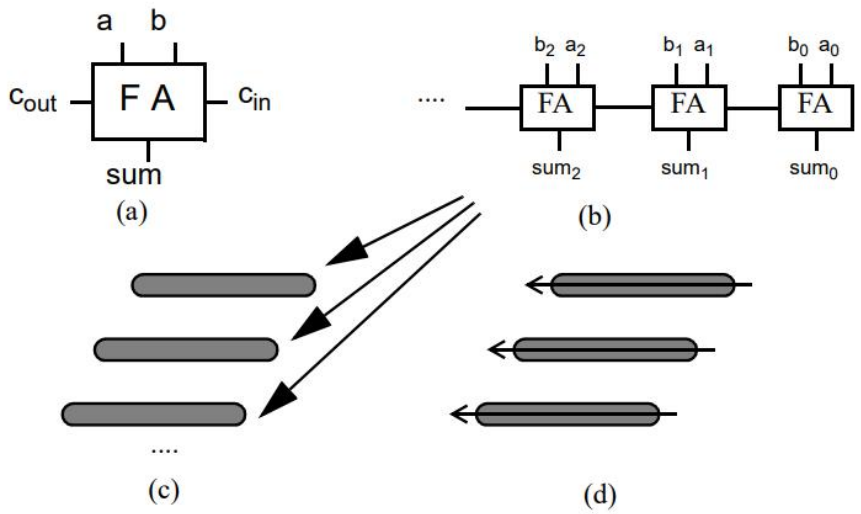


Figure 2.3. illustrates the addition of partial products (PP). (a) showcases the block diagram of a full adder. (b) presents the basic block RA (ripple adder). (c) demonstrates ripple addition employed to add copies of the multiplicand. (d) describes the basic method of ripple adders with a complexity of O(n^2) [58].

A and operation with B, afterward add C then aftereffect of A+B. Quantity of piece increments are O(2n). That's what somehow happened which involve the strategy in dim-witted design for add n moved duplicates of n-bit C.This occurs because we assume that addition operations depend on previous addition operations, where the outputs of earlier operations serve as inputs to subsequent operations. See Figure 2.3. Albeit the eventual outcome happens from joining tasks, measure of freedom exists with every activity. Think about add activity at segment using section premise; every one of the pieces in a section should be added together, alongside the convey in pieces of the past section. The postpone in working out the result in section is an element to convey in bit stream (taken from past segment) is accessible, the quantity of pieces which are using for addition as shown in figure 3 Convey store expansion use the way which add activities in discrete sections autonomously performed.

2.1.4 Array Style Reduction

Array Style Reduction (ASR) plays a pivotal role in crafting efficient digital multipliers, especially within low-power and high-performance scenarios. This review explores the importance of ASR and its ramifications on the design and enhancement of multipliers.

In this segment, we depict the most essential strategy, called cluster fractional item decrease.Putting it together, "cluster fractional item decrease" could potentially refer to a process or technique aimed at reducing the number of fractional items within a cluster. This could involve methods for consolidating or combining similar fractional values within the cluster to simplify the data set or improve computational efficiency.

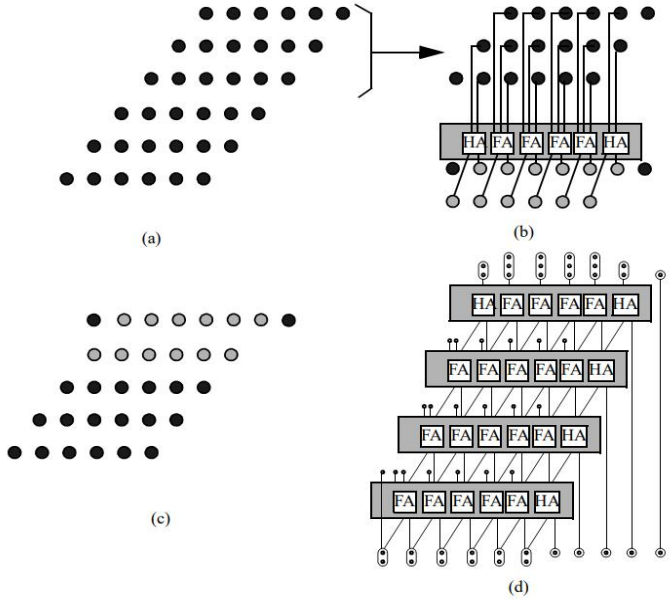


Figure 2.4. Array to reduce PP (partial product) - (a)The PP (partial product) (b) Carry save full adder to reduce 3 bit (c) resulting PPA (d) Whole [59].

array. For instance, in the Figure 2.4a, shown the PPA produced for the 6-cycle by 6-digit duplication. Consolidating consequences to expansion using leftover pieces for PPA, it obtain an outcome to shows up in Figure 2.4c. Three vectors diminished to 2 vectors. This design rehashes until the full cluster is launched as in Figure 2.4 the outstanding trademark about the cluster design is its ordinary construction [59]. It is beneficial to spread out, solitary viper block & related connections ions recreated width, profundity of cluster.

2.1.5 Wallace Tree Partial Product Reduction

Wallace Tree Partial Product Reduction stands as a technique employed in digital circuit design, specifically in crafting high-speed multipliers, aimed at efficiently diminishing the quantity of partial products generated throughout the multiplication

process. While playing out a progression of free add tasks, making a design which is conceivable has less postponement by playing out the expansion activities in equal, where conceivable [59]. For instance, in the halfway item exhibit for 6*6-digit augmentation, it convey store decreases in resemble, coming about with more modest PPA during only first stage (Figure 2.5a-c.) It rehashed, strongest design displayed in Figure 2.5d. (Last arrangement for associations are some degree muddled to draw here.)

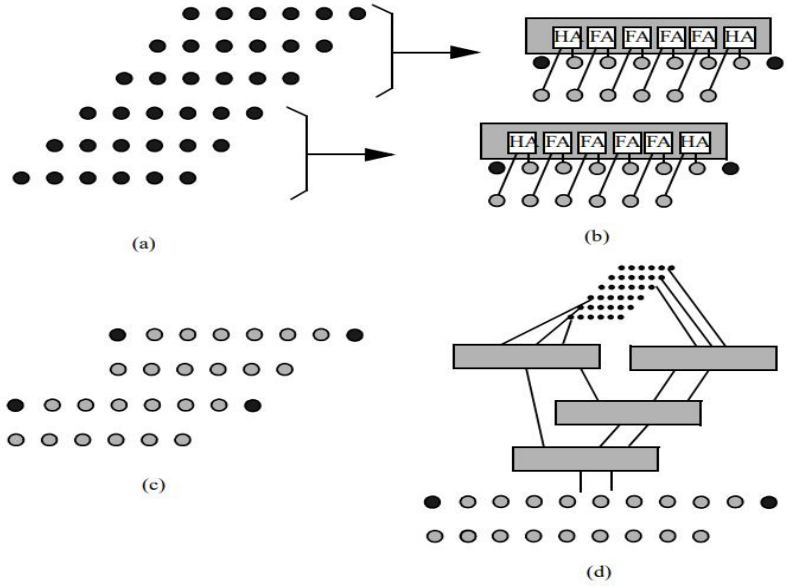


Figure 2.5 illustrates the Wallace tree partial product reduction. (a) showcases the partial product array, (b) demonstrates parallel carry-save addition, (c) presents the resulting PPA, and (d) depicts the complete Wallace tree structure [59].

For a 6-bit multiplier, this algorithm would involve grouping the partial products into sets of three and recursively reducing them until a final sum is obtained. While utilizing convey save expansion, we use 3 input piece vectors & diminish to 2 result bit. Successive convey strategy decrease quantity to piece vectors each stage one, though equal strategy can use sets of three vectors,reduce to two. Quantity of steps, subsequently the reduction consecutive activity is O(n), while equal strategy is o(log(3/2) n]. Such equal game plans of Carry save adder blocks are known as Wallace trees, consider an enormous decrease in the postponement of the halfway item decrease stage. Weakness of the trees lies in sporadic format (particularly

concerning cluster structures), coming about in possibly more noteworthy loads. Moreover, this design bring about a last addition activity of n bit width.

2.1.6 Booth Re coding to PP reduction

Booth re-coding is a strategy employed to enhance the generation of partial products in binary multiplication by leveraging patterns of adjacent bits in the multiplier. This technique reduces the quantity of partial products generated during the multiplication process, leading to improved efficiency and decreased hardware complexity.

Table 2.1 Booth Encoding

x_{2i+1}	x_{2i}	x_{2i-1}	d_i
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	2
1	0	0	-2
1	0	1	-1
1	1	0	-1
1	1	1	0

Tragically, it isn't by and large conceivable to know deduced which pieces of the multiplier will be '0'. To keep up with consensus, we should accommodate the situation when all pieces of the multiplier are '1'. It could be feasible to lessen hardware, notwithstanding, assuming biggest defer situation. A 4*4 digit cycle increase, hardware should be accommodated the situation where the multiplier is '1111'- - - bringing about a postponement of 4 phases. A significant perception is that duplicating by '1111' is equivalent to increasing as '10000', & taking away the multiplicand from the output (Result)- - - duplicating to a force basically shifting two phases. Subsequently, we have reduce 4phase to 2 phase of postponement. The encoding plan is displayed in Table 2.1, from [60]. The associations of the Stall multiplexers are displayed in Figure 2.6. As a result, the size of the PPA cluster is

reduced, as fewer shifted duplicates of the multiplicand are required in the partial product array

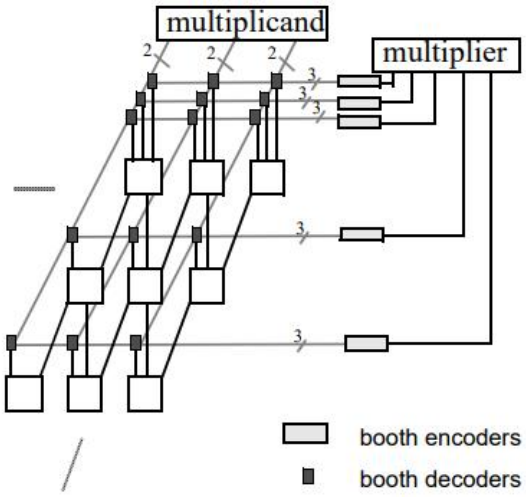


Figure. 2.6. Result, the size of the PPA cluster is reduced [60].

The quantity of these multiplicand 'duplicates' required in the halfway item exhibit relies upon how much Corner re-coding is applied. By and large, each degree of re-coding cuts the quantity of fractional item bits fifty. There is extra hardware engaged with playing out the re-coding in any case, so this improvement involves embedding confounded rationale which itself adds deferral and power utilization. By and by, Booth re-coding optimizes the multiplication process by generating partial products only when necessary, thereby reducing the overall computational load and improving efficiency in binary multiplication operations [60].

2.1.7 Final Addition of Multiplier

Typical strategy to accomplishing less postpone in multipliers is to accelerate the last expansion state. A few improvements reviewed for resulting increased velocity expansion, summed up [61]. Direct use of the plans to multiplication technique has come about different plans using rapid or reduced power . During Examinations, it is convey swell, convey skip, convey select snake method for this last expansion stage.

Ripple Adder

We know already, that wave snake very slow yet most reduced parameter power viper execution. The wave viper is of extraordinary interest on the grounds that higher speed adders frequently integrate the swell construction in small blocks (sub-blocks) of more noteworthy maximum velocity viper architecture. Thus speed furthermore, parameter characteristics of the wave viper decide general presentation to an extensive variety of viper plans. The design of wave expansion is displayed beneath in Figure 2.7. It is visible delay as a straight in quantity of expansion stage.

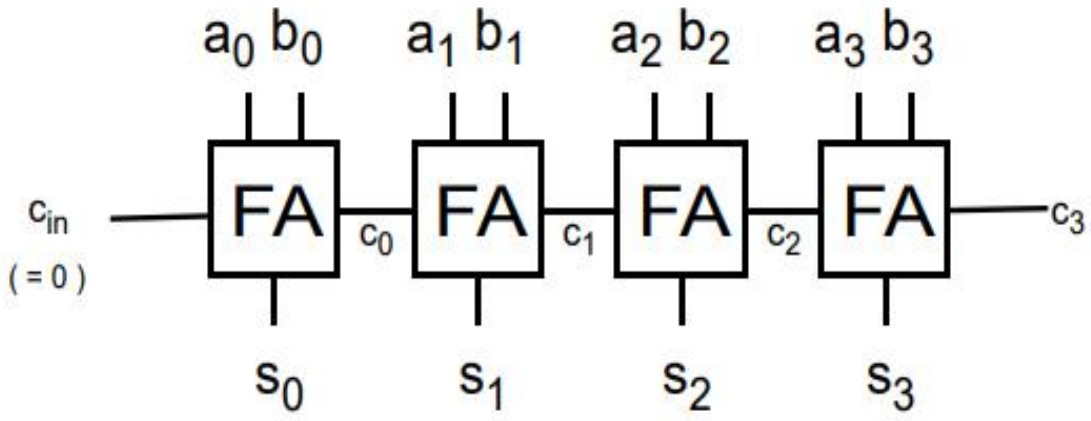


Figure 2.7. Block diagram of Ripple adder [61]

Carry-Skip Adders

A Carry Skip Adder (CSA) is a specific type of adder circuit employed in digital arithmetic operations to sum two binary numbers. Its design aims to enhance addition speed by mitigating the propagation delay linked with carry propagation. The CSA consists of multiple stages of full adders arranged in a ripple-carry manner.In each stage, there is typically a grouping of full adders interconnected in parallel. The inputs to each full adder within a stage comprise the corresponding bits from the two numbers being added and the carry-in from the preceding stage. In a conventional ripple-carry adder, the carry produced by each full adder needs to propagate through all subsequent stages before arriving at the final stage.This sequential carry propagation can introduce significant delay, limiting the speed of addition.

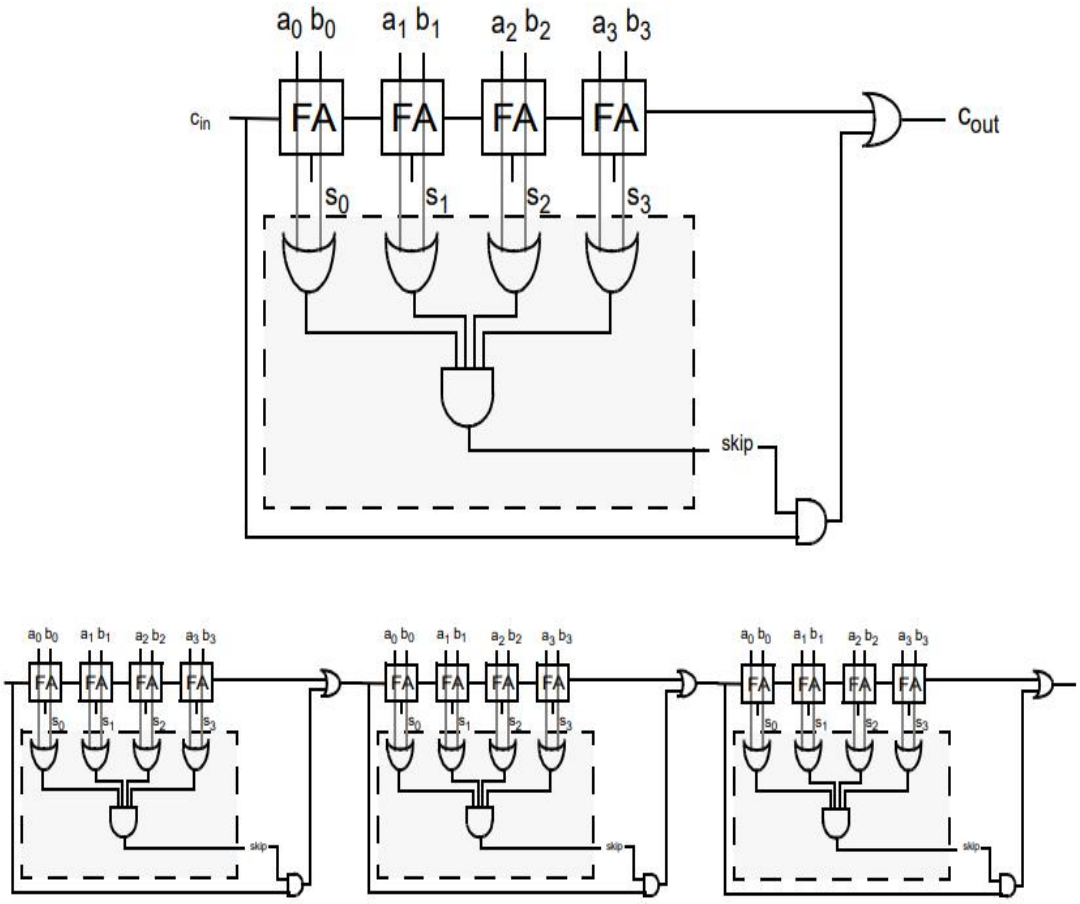


Figure 2.8. illustrates Carry Skip Adders: (a) showcasing one carry-skip block, and (b) depicting a 12-bit adder [61]

In a Carry Skip Adder, additional logic is introduced to enable carry propagation to "skip" certain stages, reducing the overall delay.Conditional logic determines whether the carry generated in a stage can be safely skipped and propagated directly to a subsequent stage without waiting for the intermediate stages.The skip logic typically involves comparing the carry signals generated in each stage to determine if they are all zero or all one.If all the carry signals in a group of consecutive stages are zero or one, indicating no carry propagation across those stages, the carry can be skipped, and the carry-in to the subsequent stage can be directly derived from the input carries.[33].Carry Skip Adders are particularly useful for large binary numbers where carry propagation delay can become a significant bottleneck.By allowing carry signals

to skip multiple stages, CSA reduces the overall propagation delay, leading to faster addition operations compared to conventional ripple-carry adders.Overall, Carry Skip Adders offer an efficient solution for high-speed addition in digital circuits by leveraging skip logic to minimize carry propagation delay.

Carry-Select Adder

A Carry-Select Adder (CSA) is a parallel adder utilized in digital circuits for summing two binary numbers. Its design aims to enhance addition speed by minimizing the critical path delay linked with carry propagation. The Carry-Select Adder comprises two distinct ripple-carry adder blocks, commonly. labeled as "fast" and "slow" adders. Each ripple-carry adder block functions autonomously and simultaneously.The "fast" adder is optimized for speed and is employed when there is no carry-in from the preceding stage. Conversely, the "slow" adder is tailored for generating the sum when a carry-in from the preceding stage exists. Both the fast and slow adders independently generate carry signals. The fast adder produces carry signals under the assumption of no carry-in from the previous stage, while the slow adder generates carry signals while considering the potential for a carry-in from the previous stage.A multiplexer (MUX) is utilized to choose the appropriate carry-out signals from the fast and slow adders based on the presence or absence of a carry-in from the previous stage. If there is no carry-in, the carry-out from the fast adder is selected due to its faster operation. Conversely, if there is a carry-in, the carry-out from the slow adder is chosen to ensure accurate summation.The Carry-Select Adder reduces the critical path delay by utilizing two parallel adders and selecting the appropriate carry-out based on the carry-in condition.It's particularly effective for addition operations where carry propagation delay is a significant bottleneck.he Carry-Select Adder provides an efficient solution for high-speed addition in digital circuits by utilizing parallelism and carry selection techniques to minimize critical path delay [61]. At a point when genuine worth of the convey in to the block is known, the right wave viper is chosen by means of a multiplexer (Figure 2.9a). A graph of deferral bit wise is displayed in Figure 2.9b.

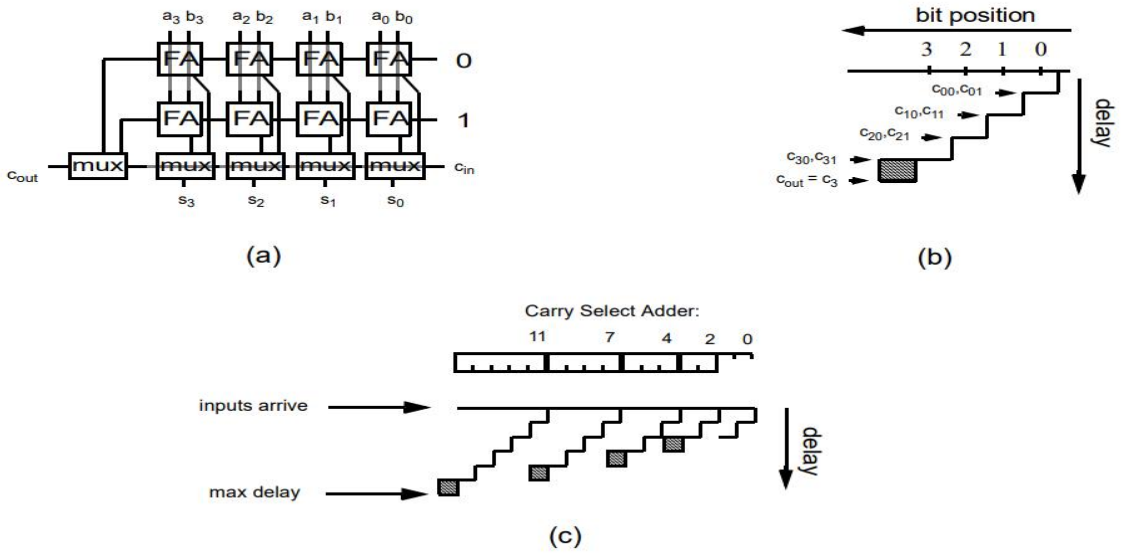


Figure 2.9. illustrates the diagram of Carry Select Adders: (a) presenting the block diagram of the adder, (b) depicting the bit-wise delay, and (c) showcasing the representation for delay [61].

2.1.8 Summary

As multiplication is a highly intricate operation, various enhancements have been devised to mitigate its delay. Speeding up the multiplier involves reducing the delay in several of its component blocks. Recent considerations of power distribution may influence which paths the multiplier delay is reduced along.

2.2 CMOS Power

CMOS (Complementary Metal-Oxide-Semiconductor) power denotes the power consumption linked with CMOS digital circuits. CMOS technology finds widespread use in integrated circuits due to its low power consumption characteristics, making it suitable for a diverse range of applications, including microprocessors, memory chips, and sensors.

2.2.1 Basics of Static and Dynamic Power

Static power, also known as leakage power, is the power consumed by CMOS circuits even when they are not switching. It arises due to sub threshold leakage currents flowing through the transistors, which increases with shrinking transistor sizes and higher operating temperatures.As feature sizes in modern CMOS technologies continue to shrink, static power consumption becomes increasingly significant. This

consumption arises due to sub threshold leakage currents flowing through transistors, particularly pronounced at higher operating temperatures.

Dynamic power consumption, in contrast, arises from the charging and discharging of capacitive elements within the circuit as it switches states. It is directly related to the switching activity and operating frequency of the circuit. The primary contributor to dynamic power consumption in CMOS circuits is the charging and discharging of the load capacitance of transistors.

Dynamic power scattering in CMOS can be portrayed by the situation in Fig. 2.9.

$$P = \alpha C V_{dd}^2 + K I V_{dd} \qquad (2.1)$$

α - activity factor. Number of transitions (per operation, e.g., in one cycle.)

C- switched capacitance.

V_{dd} - supply voltage.

KI - static current component, K- W/L of PMOS & NMOS devices, input signal slope.

Equation (2.1) represents the power consumption for CMOS. In equation alpha is the activity factor, C is the switched capacitance, V_{dd} is the supply voltage and KI is the static current for the CMOS.

2.2.2 Power Optimization Fundamentals

The power consumption of CMOS circuits is influenced by the power supply voltage. Decreasing the power supply voltage decreases dynamic power consumption, yet it might amplify the effects of process variations and delay. However, reducing the power supply voltage also decreases the dynamic range and signal-to-noise ratio of the circuit. Various techniques are employed to optimize CMOS power consumption, including:

Voltage scaling: Adjusting the power supply voltage to trade off between power consumption and performance.

Clock gating: Disabling clock signals to inactive circuit blocks to reduce dynamic power.

Pipeline optimization: Breaking down complex operations into smaller stages to reduce dynamic power. Transistor sizing involves adjusting the dimensions of transistors to strike a balance between performance and power consumption. Power gating entails completely shutting off power to inactive circuit blocks to minimize

leakage power, which is especially beneficial in reducing standby power consumption in modern integrated circuits.

2.3 Literature Review to Identify GAP

Liu et al. (2018) introduced a design for an approximate Booth multiplier and conducted an assessment of approximate logarithmic multipliers. They proposed a novel methodology for error analysis and simulation. In their synthesis of the multiplier, they utilized SDC with the NAND gate 45 nm Open Cell Library. They proposed an 8-bit multiplier and compared various parameters of the multiplier with those in the literature, showing the following parameters for the 8-bit multiplier: Power: 127.4 μW, Delay: 0.58 ns, and Area: 538.6 μm². This multiplier design was implemented in low power, error-tolerant applications [62].

Strollo et al. (2020) conducted a study focused on reviewing, comparing, and extending the use of inaccurate 4-2 compressors for approximate low-power multipliers (LPMs). He designed compressor based 8 bit low power approximate and 16 bit low power approximate multipliers. In addition, they provided a comprehensive review and comparison of various approximate 4-2 compressors based on existing literature. For synthesizing the multipliers, they employed SDC (Synopsys Design Compiler) with the 28 nm CMOS technology. Proposed work can be applicable of the Error tolerant applications like image compression and possible work include: recursive multiplier can be designed [63].

Minho Ha & Sunggu Lee (2017) proposed a design for imprecise n-bit multipliers by introducing new 4-2 inaccurate compressors and Error Recovery Modules (ERM). In their methodology, they utilized the Partial Product (PP) compression process employing truncation. The novel design of the approximate compressors for the multiplier included Error Detection Modules (EDM) and Error Recovery Modules (ERM). They used ASIC post-synthesis implementation technology for synthesizing the multipliers. They proposed an 8-bit multiplier and compared various parameters of the multiplier with existing literature, showing the following parameters for the 8-bit multiplier: Area (m²) = 510.08, Power (mW) = 0.1646, and Delay (ns) = 2.70 [64].

Liu et al. (2017) introduced a novel inaccurate Radix-4 Booth low-power multiplier tailored for error-tolerant applications, showcasing remarkable performance. Their design featured a new inaccurate 16-bit compressor based on Radix-4 Booth encoding. They conducted a thorough analysis of the error characteristics associated with their design. For synthesizing the multiplier, they leveraged SDC with the NAND Gate 45 nm Open Cell Library. They proposed a 16-bit multiplier and enhanced the parameters, achieving the following results for the 8-bit multiplier: Power (μW) = 188.0, Delay (ns) = 0.70, and Area (μm²) = 788.4. The proposed design can be used in typical error-resilient applications. Their work was implemented on an Image Processing application [65].

Yang et al. (2018) presented a very high-speed, accurate design for approximate low-power multipliers. In this design, the authors used a carry-maskable adder and compressors. The use of this methodology yielded better results. They employed an accuracy-controllable multiplier technique, where the final product is produced by a carry-mistakable adder, resulting in improved performance. They used Synopsys Design Compiler with 45nm CMOS technology at a frequency of 0.5 GHz. The authors implemented their work for an 8-bit multiplier and achieved the following parameters: Power (μW) = 180, Delay (ns) = 1.6, and Area (μm²) = 250. This multiplier was also applied to an image processing application in their work [66].

Natrajan P.B et al. (2021) introduced a work focusing on the method to encode the faulty row in the compressor's truth table (TT). This work is applicable for efficient output, high quality, low power, and error-balanced applications, such as joint group photography and image sharpening processes. In this work, a 4:2 compressor is used for a 4x4 bit multiplier, which is then scaled to 8x8 bit and 16x16 bit multipliers. The results of the introduced error-tolerant multipliers were applied to image sharpening applications [67].

Salim Ullah et al. (2021) developed an approximate signed multiplier tailored for FPGA-based systems. The proposed methodology utilized the VHDL programming language, Xilinx Vivado 17.4 software, and the Vertex-7 FPGA XC7V585T device for implementing the radix-4 Booth multiplier. Dynamic power values were

calculated using the same tool. Additionally, the study explored the optimization of area and energy efficiency of the Radix-4 signed Booth multiplier [68].

Haoran Pei et al. (2021) worked on an approximate compressor output, which was reduced to one for improved energy efficiency. Simulation results show that the proposed compressors, named UCA1 (MUL 1), UCA2 (MUL 2), and UCA3 (MUL 3), achieved a reduction in delay of 24.76%, 51.43%, and 66.67%, respectively. The reduction in power was 71.76%, 83.06%, and 93.28%, respectively, while the reduction in area was 54.02%, 79.32%, and 93.10%, respectively. This new design of the multiplier is used in applications where high speed is required, with lower power requirements, achieving an average reduction of 49.29%. This work demonstrates that compared to previous methods, the parameters have been improved, resulting in an enhanced electrical approximate multiplier [69].

Nagarajan (2022) proposed a design for a precise multiplier using inexact compressors to improve performance in digital signal processing applications. The work evaluates the performance of the proposed multiplier by comparing it with existing designs in terms of power consumption, delay, and area. Simulation results demonstrate that the proposed design achieves significant reductions in power usage and area, while maintaining acceptable error margins suitable for DSP applications. The multiplier design is particularly well-suited for error-resilient systems, offering an effective solution for power-efficient and performance-critical tasks [20].

For the thorough literature review, we analyzed the literature as shown in Table 2.2.

Table 2.2 Analysis of various Low Power Multiplier

S.no.	Title of Paper	Year	Indexing of journal	Concept/ circuit used	Methods/ work	Applications
1.	Design of precise multiplier using inexact compressor for digital signal processing [20]	2022	Comput. Syst. Sci. Eng. Vol.42 (2)	Inexact compressor within the multiplication process	proposed design achieves significant reductions in power usage and area, while maintaining acceptable error margins.	Error-resilient systems,power-efficient and performance-critical tasks

2.	To Design and Evaluation of Approximate Logarithmic Multipliers[70].	2018	IEEE Trans. Vol.65(9).	Approximate Booth multipliers	Error analysis and simulation	Low Power Error-Tolerant Applicat ions
3.	Review on Comparison and Extension of Approximate 4-2 Compressors for Low- Power Approximate Multipliers [71].	2020	IEEE Trans. Vol.67 (9)	To design 8 × 8 and 16 × 16 multiplie rs Compres sor.	Comprehensiv e survey and Comparison of approximate 4-2 compressors previously proposed In literature.	In typical error- resilient applications Work Implemented on Image Processing
4.	Design and Analysis of Approximate Compressors for Multiplication [72].	2015	IEEE Trans. Vol.64 (4)	analyzed and designof two new approximate 4-2 compressors	Extensive simulation resultsare provided withthe Dadda Multiplier	application of the approximate multipliers to image processing is presented
5.	A Novel Design for Multipliers with Approximate 4-2 Compressors and Error Recovery Modules. [73].	2017	IEEE System letters. Vol.10 (1)	Approximate n bit multipliers using the proposed 4-2 compressor and errorrecovery module.	Partial product accumulation process using truncation and proposed approximate compressors for an 8-bit approximate multiplier with errordetection module and error recovery module	Error-resilient applications, image processing application, image sharpening, has been used.
6.	Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing [74].	2017	IEEE Trans. Vol.66 (8)	Approximate 16 bit Booth multipliers are designed based on approximate Radix-4 modified Booth encoding (MBE).	Error characteristics are analyzed With respect to the approximation factor that is related to the inexact bit width of the Booth multipliers	In typical error- resilient applications Work Implimented on Image Processing
7.	A Low-Power High-Speed Accuracy-Controllable Approximate Multiplier Design [75].	2018	Asia & South Pacific design conferen ce	Multiplier design is implemented by employing the carrymaskable adder & compress or	Accuracy-controllable multiplier whosefinal product is generated By a carry-maskable adder	In Image processing application Demonstrate that the quality of the processed images can Be controlled.

8.	A library of approximate adders and multipliers Called evoapprox**8b** [76].	2017	Design, Automation & Test in Europe Conference & Exhibition	Library contains 430 nondominated 8-bit approximate adders created from 13 conventional Adders and 471 non-dominated 8-bit approximate multipliers Created.	A multi-objective Cartesian genetic Programming. Evoapprox 8b library provides Verilog, Matlab and C models	It can be utilized as building Blocks of more complex approximate adders and multipliers. Image and video processing
9.	Approximate Computing For Low Power And Security in the Internet of Things [77].	2017	IEEE Vol.50 (6)	Approximate computing With information hiding with 32 bit implementation	It utilizes the energy Efficiency of approximate computing And introduces tolerable error	To increases the security of iot devices by hiding info. for IP Watermarking, digital fingerprinting, & lightweight encryption.
10.	A Design of Power and Area Efficient Approximate Multipliers [78].	2017	IEEE Trans. Vol.25 (8)	Approximate half- adder, full-adder, and 4-2 compressor are proposed to reduce remaining partial products	Reduction in areaand power consumption compared with Exact designs	Proposed multipliersIs evaluated with an image processing application
11.	An energy efficient approximate multiplier design [79].	2017	Elsevier Vol.63 (1)	AQ-LETAM (for different output accuracy levels), and Baugh–Wooley multiplier are compared	Proposed multiplication algorithm for 32 bit multiplier	Implemented on Image Processing
12.	Implementation of low-power approximate Compressor–based multiplier for cognitive communication Systems [80].	2019	Wellay Online Liberary. Special Issue	This multiplier is tominimize the shift and add operation,	Compressor-based approximate multiplier for performance Comparison, sincethe algorithm will reduce the number of logic elements,	This architecture is implemented In CYCLONE IV FPGA with 512 × 512 and 640 × 480 resolutions of Images.
13.	A Rounding-Based Approximate Multiplier for High-Speed yet Energy Efficient Digital Signal Processing [81].	2019	International Journal of Research Vol.8(1)	Design and Hardware Implimentation of proposed MROBA Multiplier	BAM Approximation method to the Conventional modified Booth multiplier, an approximatesigned Booth multiplier was presented	Energy Efficient Digital Signal Processing. Studied in 2 image process applications, i.e., image sharpeningand smoothing

14.	Modified Rounding Based Approximate Multiplier (MROBA) and MAC Unit Design for Digital Signal Processing [82].	2018	IJPAM Vol.118 (8)	Three hardware implementations are Proposed in which one implementation for unsigned And two for signed operations	Algorithm Design and Hardware implementation of the proposed modified Rounding based approximate multiplier	DSP Applications. In the design of multiplier and accumulate unit (MAC) Multipliers
15.	Low-Power Approximate Multipliers Using Encoded Partial Products and Approximate Compressors [83].	2018	IEEE Journal Vol.8 (3)	Two 4×4 multipliers designed with different accuracies and then are used as building blocks for scaling up to 16×16 and 32×32 multipliers	Hardware implimentation and error performance metrics of the proposed and other approximate multipliers. Partial Product accumulation and scalling used.	Image sharpening and joint photographic experts group (JPEG) applications. *MIMO Systems in wireless Communication systems.*
16	Low Power Approximate DCT Architecture for HEVC Standard [84].	2017	Design, Automation & Test in Europe Conference.	Proposed multiplierless 4-input DCT implementations	Design of approximate adders and subtractors . Approximate adders and subtractors that were obtained using Genetic programming and error analysis.	Video processing. IOT Applications.
17	Optimization of Constant Matrix Multiplication With Low Power and High Throughput [85].	2017	IEEE Trans. Vol.66 (12)	Constant matrix multiplication with 2 X 2 Matrix	A) reducing the adder depth (AD) Leads to a reduced power consumption B) pipeline resources have To be considered during optimization to enhance throughput without Wasting area.	In DSP applications. Constant rotators as used in the butterflies of Fast Fourier transform (FFT) implementations
18	A Low-Power Convolutional Neural Network Face Recognition Processor and a CIS Integrated With lways-on Face Detector [86].	2018	IEEE Journal Vol.53 (1)	The hybrid use of analog processing unit And digital processing unit is proposed	Analog–digital Hybrid Haar-like FD is proposed To improve the energy efficiency.	For the User authentication in smart devices. Image sensors.

19	A 90-nm CMOS Low-Energy Dual-Channel Serial/Parallel Multiplier [87].	2019	International Conf. ACCS. IEEE	Serial/parallel scheme	Comparative studyof different multipliers and simulation is accomplished at 90nm CMOS technology. Decreasing power consumption.	In low power digitaldesign applications for less hardware cost and energy
20	Capacitance Super MultiplierFor Sub-Hertz Low-Pass Integrated Filters [88].	2017	IEEE Trans. Vol.65 (3)	Capacitor super multiplier technique	Simulations at the transistor level using a 65-nm CMOS Technology	Biomedical area, envisage Integrated architectures with filters tunable at veryLow frequencies
21	Efficient FPGA Implementation of Low-Complexity Systolic Karatsuba Multiplier Over *GF(2m)* Based on NIST Polynomials [89].	2017	IEEE Trans. Vol. 64(7)	Karatsuba algorithm (KA)-based digit- serial multiplier	Algorithm and Architecture for efficient FPGA implementation ofKA-based Digit-serial systolic multiplier over GF(2m)	Wearable devicesand Deeply embedded systems.
22	A Low Error Energy- Efficient Fixed-Width BoothMultiplier with Sign-Digit-Based Conditional Probability Estimation [90].	2016	IEEE Trans. Vol.65 (2)	Booth encoded sign-digit-based conditional probability estimation approach is proposed	Circuit implementation which reduces the delay time, as wellas power dissipation. Simulation for best computation accuracy.	Modern energy-efficient multimedia and digital signal processing systems
23	A Low-Cost Voltage Equalizer Based on Wireless Power Transfer and Voltage Multiplier [91].	2017	IEEE Trans. vol.65 (7)	A single-switch Voltage equalizer without feedback control used for voltage Multiplier.	Characteristics of a multiple-Output VM are analytically discussed and usedto develop An equivalent one-output VM mode and parameter design.	Wearable devices, cellphones, household Appliances, and even electric vehicles
24	Design of low error fixed-width modified Booth multiplier [92].	2004	IEEE Trans. vol. 12(5)	error compensation method for a modified Booth fixed-width multiplier	proposed method leads to up to 35% reduction in area and power consumption	For applications of Low Power Error-Tolerant

25	New Metrics for the Reliability of Approximateand Probabilistic Adders [93].	2013	IEEE Trans.vo l62(9),	new metrics are proposed for evaluating the reliability as well asthe power efficiencyof approximate and probabilistic adders	proposed several new metrics for evaluating approximate and probabilistic adderswith respect to their reliability and power efficiency	proposed metrics may also be useful inassessing other arithmetic circuits for inexact computing and/or fault-tolerant designs in nano computing app lications.
26	Non-Volatile Approximate Arithmetic Circuits using Scalable Hybrid Spin-CMOS Majority Gates [94].	2021	IEEE Trans.vo l68(3),	proposed 4-2 compressors show shorter critical path delays and lower energy consumption	propose to use spintronic deviceswith near-zero leakage power and non-volatility as key components in arithmetic circuits	For applictatio ns of error-resilient
27	Design and Performance Analysis of Rounding Approximate Multiplier for Signal rocessing Applications [95].	2021	SCOPUS	Minimize the powerutilization factor and also reduces errors	Propose the utilization of power	For applictatio ns of error-resilient
28	Approximate radix-8 Booth multiplier for low power and high speed applications [96].	2020	SCI	Reduces the area and energy of multiplier	Propose area reduction methods	For applictatio ns of error-resilient

2.4 Literature Gap

The reviewed literature highlights a wide spectrum of multiplier architectures and optimization techniques developed for enhancing performance, reducing power, and optimizing area in digital systems, particularly in DSP and MAC unit applications. Moreover, approximate logarithmic multipliers and dynamic precision control architectures were introduced to support high flexibility in power-accuracy trade-offs, especially for low-power embedded systems and IoT devices. Despite these advancements, certain research gaps persist:

1. Unified Optimization Framework: While individual studies focus on optimizing either area, power, or delay, a comprehensive framework addressing all three parameters holistically for different use-case scenarios is still lacking.

2. Lack of Application-Oriented Architectures: Many designs do not extend their implementation to real-world DSP or edge computing scenarios. There is a gap in integrating approximate multiplier designs directly into application-specific processors, especially for image and signal processing tasks.
3. Error Metrics and Quality Trade-off Control: While some works explore approximate computing, dynamic error control or output quality tunability remains underexplored. Most existing designs offer static approximations with no dynamic configurability.
4. Compressor-Based Multipliers: Although compressors like 4:2 and 8:2 are used for delay reduction, in-depth exploration of inexact compressors in high-speed multiplier architectures is limited.
5. Comparison Under Uniform Conditions: The results across studies use varied technology nodes, simulation tools, and benchmarks, making cross-comparison difficult. There's a lack of standardized evaluation metrics or platforms.
6. Power-Scalable Architectures: Few architectures are able to dynamically scale power consumption based on real-time workload requirements, a crucial feature for modern energy-constrained applications.
7. Integration and Synthesis Constraints: Most designs are validated only via simulation, and very few are tested through complete synthesis and FPGA/ASIC implementation, limiting their practical verification and deployment.
8. Post-Layout Evaluation and Reliability: Thermal effects, aging, and process variations are rarely addressed in multiplier designs, which are important for real-world robust implementations.

In conclusion, while existing literature provides diverse approaches to low-power and approximate multiplier design, this thesis identifies the need for a power-efficient, delay-optimized, and area-conscious multiplier architecture that incorporates inexact compressor-based designs, supports dynamic approximation control, and is suitable for real-time DSP applications—a gap that this research aims to fill.

2.5 Summary

In this section, we provide an overview of multiplier operations, along with the essential power distribution characteristics in CMOS. While distortions and parasitic

currents are inherent in CMOS implementations of multipliers, our aim is to mitigate these effects while performing the operation. Design methodologies have increasingly focused on power reduction in recent times, although the overarching goal is to achieve power efficiency without compromising on delay, a more challenging task. In subsequent sections, we detail the analysis and design of multipliers for low power, utilizing multiplier architectures to support our conclusions. Special attention is given to physical design characteristics, which can influence decisions made at the logic level.

Parameter Establishment:

Establishing operating parameters for the categorization of approximate low-power multipliers involves defining criteria that can be used to classify and compare different multiplier designs based on their power efficiency and approximation accuracy. Here are some key parameters that can be considered:

1. **Power Consumption:**
- Average power consumption during operation.
- Dynamic power consumption under different input conditions.
- Static power consumption when idle.
- Power efficiency metrics such as energy per operation or energy-delay product.

2. **Area Efficiency:**
- Utilization of hardware resources such as FPGA slices or ASIC area.
- Silicon area occupied by the multiplier circuitry.
- Efficiency in terms of area utilized per operation or per bit precision.

3. **Delay and Throughput:**
- Propagation delay of the multiplier circuit.
- Critical path delay and worst-case delay under different input conditions.
- Through put, or the number of operations per unit time.
- Trade-offs between delay, throughput, and accuracy.

4. **Accuracy and Error Metrics:**
- Approximation error introduced by the multiplier design.
- Mean Square Error (MSE) or other error metrics compared to an ideal multiplier.

- Error distribution and worst-case error scenarios.

5. **Design Complexity:**

- Complexity of the multiplier architecture.
- Number of components or stages in the multiplier circuit.
- Complexity of approximation techniques used.

6. **Robustness and Reliability:**

- Sensitivity to process variations and environmental factors.
- Robustness to noise and jitters.
- Reliability in terms of error resilience and fault tolerance.

7. **Flexibility and Configuration:**

- Configuration of the multiplier for different precision requirements.
- Flexibility in adapting to different application scenarios.
- Ability to adjust approximation levels based on power constraints or performance requirements.

8. **Application-specific Metrics:**

- Relevance and performance in specific application area as image processing, machine learning, or signal processing.
- Suitability for low-power embedded systems or battery-powered devices.

By defining and quantifying these parameters, researchers can systematically evaluate and compare different approximate low-power multiplier designs, enabling the classification of multipliers into categories based on their performance characteristics. This categorization can aid in selecting the most suitable multiplier design for specific applications and design constraints.

CHAPTER 3

Optimization Techniques and Proposed Methodology

3.1 Introduction

In this section, we explore optimization techniques and a proposed methodology aimed at improving the performance of the 4:2 inexact compressor-based 8-bit Dadda multiplier, focusing on its application in image processing. The objectives set for this endeavor underscore the significance of achieving efficiency in terms of area utilization, delay reduction, and power optimization. To fulfill these objectives, a systematic approach to optimization is necessary, leveraging a combination of algorithmic enhancements and architectural refinements. The following introduction sets the stage for the exploration of optimization techniques and the proposed methodology.

In the domain of digital signal processing (DSP), the creation and deployment of efficient multipliers are paramount given their ubiquitous role in computational tasks. In this regard, the 4:2 inexact compressor-based 8-bit Dadda multiplier emerges as a promising contender, holding the promise of benefits such as area efficiency, minimized delay, and optimized power utilization However, realizing these advantages requires a comprehensive exploration of optimization techniques tailored to the unique characteristics of the multiplier architecture.

This section focuses on elucidating a structured approach to optimization, encompassing both algorithmic innovations and architectural optimizations. Our goal is twofold: to improve the performance of the multiplier design while preserving its low-power features, and to enable its smooth integration into DSP processors for image processing applications.

To achieve these goals, we will first establish operating parameters that delineate the criteria for categorizing an approximate low-power multiplier. Subsequently, we will delve into the design of an algorithm tailored specifically for low-power operation, leveraging insights gleaned from the established parameters. Our approach will then

pivot towards optimizing power consumption through judicious utilization of synthesis and implementation tools offered by Xilinx and Cadence in their 45nm process technology. Finally, we will devise an architecture that embodies the synergy between low-power design principles and the computational requirements of image processing, ensuring compatibility and efficiency in real-world deployment scenarios.

By delineating the optimization techniques and proposed methodology in this section, we pave the way for a systematic exploration of strategies aimed at maximizing the performance and efficacy of the 4:2 inexact compressor-based 8-bit Dadda multiplier in image processing applications [69]. This introduction sets the context for the subsequent discussion on optimization techniques and proposed methodologies, providing a road map for achieving the objectives of the proposed work.

3.2 Operating Parameters for Categorization

Before delving into the optimization techniques and proposed methodology, it's crucial to establish the operating parameters that will serve as the foundation for categorizing the approximate low-power multiplier. Here's an outline of the operating parameters for categorization, which will serve as a framework for evaluating the performance and efficiency of the multiplier design.

1. **Area Efficiency**
   - Define metrics to quantify the area utilization of the multiplier design.
   - Measure the number of logic elements, flip-flops, and other resources consumed by the multiplier.
   - Establish criteria for categorizing the multiplier based on its area efficiency, such as resource utilization per bit or per operation.

2. **Delay Characteristics:**
   - Identify parameters related to delay characteristics, including propagation delay and critical path delay.
   - Measure the delay introduced by the multiplier during operation.
   - Establish thresholds for delay characteristics to differentiate between fast and slow multiplier designs.

3. **Power Consumption:**

- Define metrics for quantifying power consumption, including dynamic and static power.
- Measure the power consumption of the multiplier design under various operating conditions using different tools also.
- Establish criteria for categorizing the multiplier based on its power efficiency, such as Power consumption per operation or per unit of area.

4. **Accuracy and Precision:**

- Assess the accuracy and precision of the multiplier in delivering correct results.
- Define metrics for evaluating the accuracy of multiplication operations performed by the multiplier.
- Establish criteria for categorizing the multiplier based on its accuracy and precision in approximation.

5. **Performance Trade-offs:**

- Consider the trade-offs between area, delay, power, and accuracy in the multiplier design.
- Evaluate how different optimization techniques impact these parameters and their trade-offs.
- Establish guidelines for categorizing the multiplier based on the balance of performance parameters.

6. **Scalability and Flexibility:**

- Assess the scalability and flexibility of the multiplier design for different applications and requirements.
- Define criteria for categorizing the multiplier based on its adaptability to varying computational needs and constraints.

By establishing these operating parameters, we can systematically evaluate and categorize the 4:2 inexact compressor-based 8-bit Dadda multiplier designs based on their performance characteristics. These parameters will guide the optimization process and help in achieving the objectives of designing an efficient low-power multiplier for FPGA implementation in image processing applications.

3.3 Algorithm and Methodology

In the Optimization Techniques and Proposed Methodology section, the algorithm design is central, serving as the foundation for developing the low-power multiplier. The proposed methodology is crafted to meet the core objectives of the research. To successfully achieve these objectives, it is essential to adopt an appropriate algorithm supported by the right set of tools.

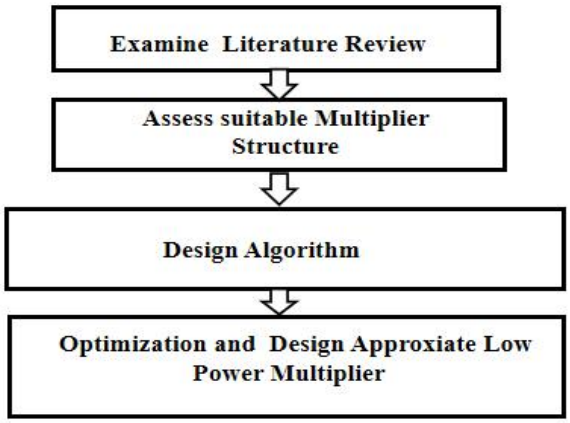


Figure 3.1 Process Flow Chart

Phase 1: Examine the existing literature to identify possible options for improving the performance of DSP processor of a low power multiplier.

Phase 2: Study various multiplication schemes to assess the suitability of multiplier structure. Power Optimization for proposed Multiplier
Phase 3: Algorithm design of multiplier for reducing parameters and conduct of experiments.

Phase 4: Optimization of Low Power Multiplier in presence of highly correlated inputs through experiments.

Phase 5: Establish design for reducing total power for the DSP and validate through experiments.
Phase 6: Refine the design based on the results obtained.

Phase 7: Conclusion, dissemination of knowledge through conference/ journal publication and thesis writing.

3.3.1 Accurate 4:2 Compressor:

Exact 4:2 compressor has four inputs and one carry input. It has two outputs Sum

and carries and one carry-out pin as shown in Figure. 3.2. Figure 3.3 shows the basic structure of a 4:2 compressor using two full adders. Compressors are used to speed up the parallel multipliers.

This means with the help of two full adder circuits a 4:2 compressor can be built. The output for the compressor circuit can be represented by the logical equations as follows:

$$\text{Sum} = A_1 * A_2 * A_3 * A_4 * C_{in}, \quad (3.1)$$

$$\text{Cout} = (A_1 * A_2)A_3 + (A_1 * A_2)^{\times}A_1, \quad (3.2)$$

$$\text{Carry} = (A_1 * A_2 * A_3 * A_4)C_{in} + (A_1 * A_2 * A_3 * A_4)^{\times}A_4. \quad (3.3)$$

A new imprecise 4:2 compressor was developed for use in multipliers used in image processing applications. In addition to output values, the pattern

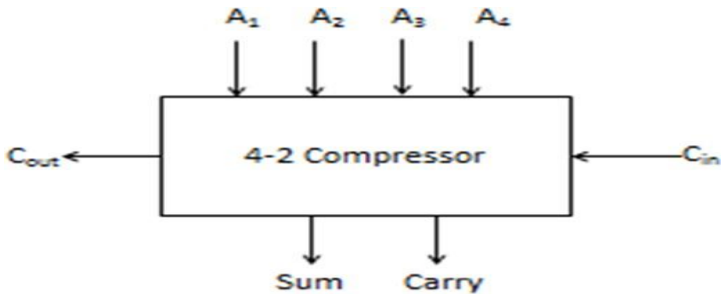


Figure. 3.2. Basic block diagram of 4–2 compressor.

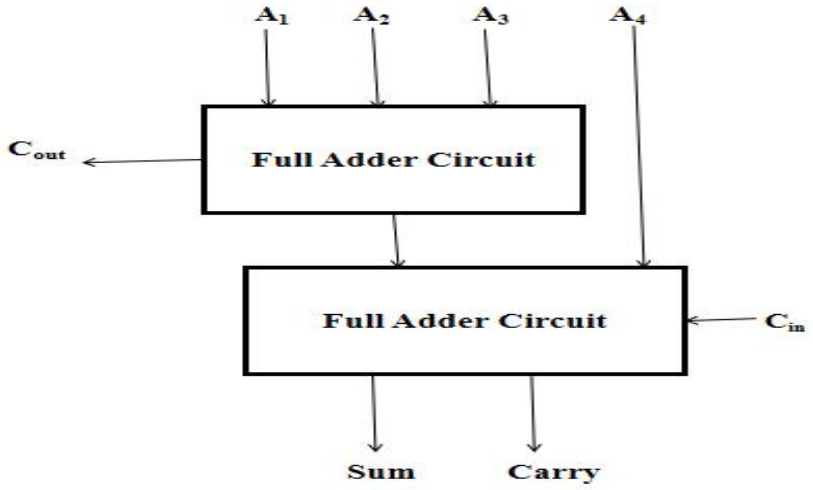


Figure. 3.3. Basic block diagram of 4:2 compressor using two full adder

distribution is also considered to synthesize the 4:2 compressor in an imprecise style. The proposed imprecise 4:2 compressor can reduce power consumption and delay by 56% and 39%, respectively, compared to a precise compressor. Simulation results indicate that the proposed imprecise 4:2 compressor can achieve a 33% improvement in power consumption and a 30% improvement in delay compared to a precise multiplier. The performance of Design 1 was compared to that of a precise 4:2 compressor. However, Design 1 sacrificed accuracy to achieve a lower cost and critical path. The error rate, which is the probability of incorrect outputs, can be calculated using the equation 3.4 .

Table 3.1. Comparison of 4:2 compressor

Design	**Critical Path**	**Error rate**
Precise 4:2 compressor	Three XOR	**0**
Design 1	One XNOR + two NOR	**37.5%**
Design 2	One XNOR + one OR	**25%**
Approximate 4:2 compressors	AND–OR complex compound gate	**25%**

The approximate 4:2 compressors have 4-input, 2-output compressors utilized in image processing applications. These approximate compressors offer reduced costs and shorter delays compared to precise compressors, albeit with higher error rates. Specifically, the approximate 4:2 compressor has an error rate of 25%, but it incurs higher costs and a longer critical path.

$$\text{Error Rate} = \frac{\text{Number of Incorrect Output}}{\text{Total Number of Output}}. \qquad (3.4)$$

In contrast, the approximate 4:2 compressor features the lowest cost and shortest critical path, though it also has the highest error rate [20]. Despite the cost similarity between the compressors, the compressor exhibits lower error rates and shorter critical paths [21]. This evaluates the proposed design against these compressors, as well as the precise 4:2 compressor, demonstrating its effectiveness.

3.3.2 Approximate Compressor:

Designing a low-power Dadda multiplier involves several key components, one of which is the 4-2 compressor. The 4-2 compressor is an essential building block that helps reduce the number of partial products generated during multiplication. Here's a detailed overview of the 4-2 compressor and its role in a low-power Dadda multiplier design.

A 4-2 compressor takes four input bits and produces two output bits, along with a carry-out bit. This is particularly useful in binary multiplication, where you often have to sum multiple bits, and a compressor can help reduce the number of bits that need to be summed at each stage.

3.3.3 Structure of the 4-2 Compressor

The circuit can be implemented using basic logic gates, and typically consists of:

1. **Full Adder Components**: The 4-2 compressor can be built using two full adders to process the inputs. The first full adder sums three of the inputs and generates an intermediate sum and carry, while the second adder combines the carry and the fourth input.
2. **Logic Gates**: Using AND, OR, and XOR gates, the necessary logic can be synthesized.

3.**Minimized Logic**: For low-power design, it's crucial to minimize the number of transistors and gates. Techniques like gate sizing and the use of transmission gates can help in reducing the power consumption.

3.3.4 Truth Table for the 4-2 Compressor

Table 3.2 shows the Truth table for the 4-2 Compressor in which we take four input variable [19]. Hence 16 combination for the inputs are presented and sum and carry as output presented for the 4-2 compressor.

For a 4-2 compressor, the truth table can be defined as follows:

Table 3.2 Truth Table for the 4-2 Compressor [19]

Input Bits	Sum (S)	Carry (c)
0 0 0 0	0 0	0
0 0 0 1	0 1	0
0 0 1 0	0 1	0
0 0 1 1	1 0	0
0 1 0 0	0 1	0
0 1 0 1	1 0	0
0 1 1 0	1 0	0
0 1 1 1	1 1	0
1 0 0 0	0 1	0
1 0 0 1	1 0	0
1 0 1 0	1 0	0
1 0 1 1	1 1	0
1 1 0 0	1 0	0
1 1 0 1	1 1	0
1 1 1 0	1 1	0
1 1 1 1	1 1	1

3.3.5 Compressor Adder

The Speculative and Almost Correct Adder (ACA) The so-called almost correct adder (ACA) is based on the speculative adder design. The ACA utilizes insufficient information, i.e., k LSBs for predicting the sum of each bit in an n-bit adder (n>k). Four bits (i.e., k = 4) are used to calculate each bit in the sum of an n-bit adder. The design is based on the observation that the carry propagation chain is usually shorter than n, i.e., in practice, the truncation of the chain up to some length has a very low probability to be erroneous. The Equal Segmentation Adder (ESA) A dynamic segmentation and error compensation (DSEC) scheme is use for an approximate adder design. This approximate adder consists of several sub-adders of different sizes divided from an n-bit adder; each of the sub-adders operates in parallel and has a truncated carry input. For convenience, but with no loss in correctness, sub-adders of

equal size are considered in this thesis. Moreover the error compensation part is neglected because the focus of this thesis is on analyzing the approximation [41].

Accuracy-Configurable Approximate Adder:

As accuracy can be configured during runtime by changing the circuit structure, a better tradeoff of accuracy versus performance and power can be achieved.

The Dithering Adder :

The dithering adder starts by dividing a multiple-bit adder into two sub-adders, and then uses a "Dither Control" signal to configure an upper or lower bound of the sum, resulting in a smaller overall error variance.

3.3.6 Implementation in Dadda Multiplier

The Dadda multiplier is a type of multiplier used in digital circuits, which employs a tree structure to reduce the number of partial products generated during multiplication. One of the key components in the Dadda multiplier is the 4-2 compressor, which helps to efficiently reduce multiple bits into fewer bits, thereby treamlining the overall multiplication process.A 4-2 compressor takes four input bits and compresses them into two output bits, along with a carry bit. The operation can be represented mathematically as follows:Gather the four input bits. For example, let's say we have inputs A0,A1,A2,A3 A_0, A_1, A_2, A_3 A0 ,A1 ,A2 ,A3 .Ensure that the outputs S0,S1,S_0, S_1,S0 ,S1 , and CCC behave as expected for all possible input combinations. This can be done using a truth table.The 4-2 compressor is crucial in the Dadda multiplier's design as it allows for a compact representation of the intermediate products. By compressing four bits into two bits and a carry, it reduces the complexity of the multiplication operation significantly. The careful design of the circuit, following the above steps, ensures the efficient processing of binary multiplication in digital systems. Create a circuit diagram that shows how these components are connected. This will typically involve: Four input lines feeding into a series of AND and XOR gates.Additional OR gates to combine carry outputs.Compute the first sum output using the XOR operation on all four inputs. For the second sum output, you can calculate it by implementing the necessary AND gates followed by the XOR gate.

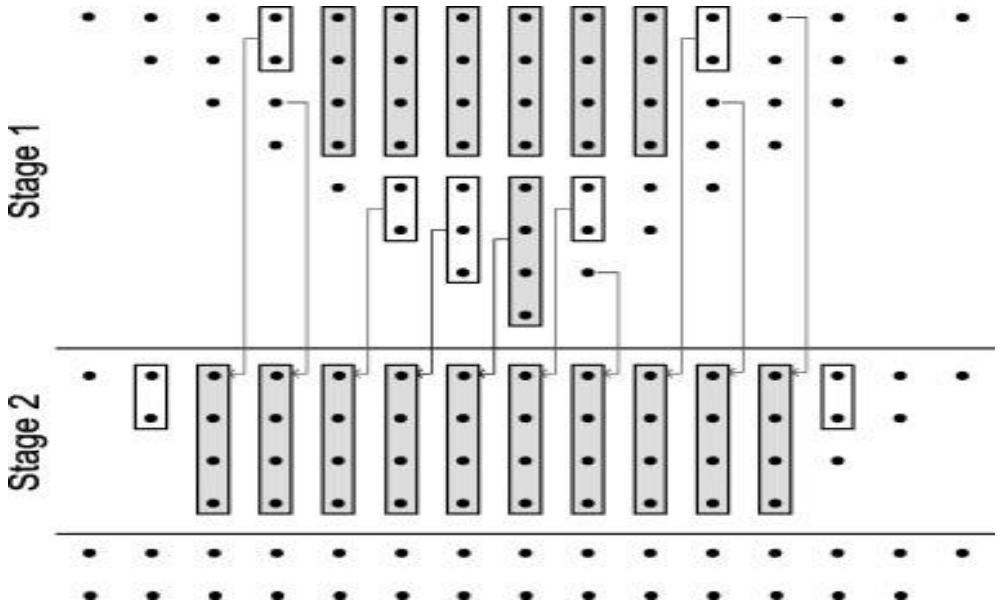


Figure 3.4 Design of Low power multiplier using Compressors

In a Dadda multiplier, the partial products are generated first, as shown in figure 3.4 and then the 4-2 compressors are employed to combine these products:

1. **Partial Product Generation**: For an N-bit multiplier, N partial products are generated, which can be represented as rows of bits.
2. **Reduction Tree**: The Dadda multiplier uses a tree structure to reduce the number of partial products. At each stage, multiple 4-2 compressors are used to compress the bits down to fewer rows.
3. **Final Addition**: After several stages of compression, a simple adder (like a carry-lookahead adder or a final ripple carry adder) is used to sum the last few rows to get the final product.

3.4 Power Optimization Techniques

In this proposed work several optimization techniques are applied across circuit design, error control, and multiplier architecture [69]. Here's a breakdown of the key optimization techniques used:

1. Logic Simplification and Output Reduction Technique: The number of outputs of the approximate 4-2 compressors (UCAC1, UCAC2, UCAC3) is reduced to one instead of the conventional three (sum, carry, cout) [69] .

Benefit: Significantly reduces gate count, leading to lower power, area, and delay.Example: UCAC3 uses a single OR gate, which is extremely efficient in terms of logic complexity.

2. Error Compensation Characteristic:Technique: The compensation characteristic of addition is used to offset negative error distances (EDs) in one compressor with positive corrections in higher significant bits.

Implementation: A correcting bit of constant logic '1' is added to the most significant bit (MSB) of a compressor chain (CC) to cancel out accumulated negative errors.

Result: This allows a chain of approximate compressors to maintain a balanced and bounded error profile.

3. Error-Correcting Module (ECM): Technique: The ECM detects input patterns (e.g., when all inputs are zero – the 0o case) where the constant correcting bit would introduce unnecessary error, and disables it by outputting logic '0'.

Benefit: Significantly improves error metrics like Mean Error (ME) and Normalized Mean Square Error (NMSE).

Impact: Makes designs like MUL2 and MUL3 better performers in terms of error resilience without drastically increasing complexity [104].

4. Partial Approximation Strategy:

Technique: Only N-1 least significant bits in the multiplier use approximate compressors; the most significant part uses exact compressors.

Reasoning: Errors in higher-order bits have more weight, so exactness is preserved where it matters most.

Trade-off: Balances error performance and energy efficiency.

5. Use of Error Metrics for Design Evaluation:

Metrics Used:

Error Rate (ER)

Error Distance (ED)

Mean Square Error (MSE)

Mean Error (ME)

Maximum Absolute Error (MAE)

Peak Signal-to-Noise Ratio (PSNR)

Optimization Goal: Guide the design trade-offs to minimize physical cost while keeping errors within acceptable bounds, particularly for image processing tasks.

6. Application-Level Evaluation:

Technique: Validate the designs using image multiplication, where human visual tolerance allows some error.

Optimization Insight: Real-world PSNR results (all >60dB for MUL2 and MUL3) confirm that approximate multipliers are viable for perception-tolerant applications.

7. Synthesis and Simulation Optimization:

Tools: Xilinx and Cadence (45-nm CMOS technology) using both slow and fast process corners, working at 100 MHz and 1V supply voltage of 1 V [104].

Purpose: Evaluate area, delay, and power using standard cell libraries for realistic benchmarks.

3.5 Error Correcting Module (ECM)

Incorporating an error-correcting module helps detect and correct potential errors in the multiplication process, ensuring the reliability of the results. The following approaches can be considered:

a. Parity Checks

- **Simple Parity**: Add a parity bit to the outputs of the compressor. This allows for single-bit error detection in the sum and carry outputs.
- **Implementation**: Use an XOR gate to compute the parity of the sum and carry outputs.

b. Hamming Code

- **Hamming Code**: Use Hamming codes for correcting single-bit errors. The 7-bit Hamming code can be used to encode the 4-bit outputs of the compressor.
- **Implementation**: Extend the 4-2 compressor output (2 bits for sum, 1 carry) into a 7-bit code-word. The additional bits are used to store parity information.

3.6 Circuit Design Considerations

a. Low-Power Design Techniques

1. **Reduced Voltage Swing**: Implement low-swing logic to minimize power consumption.
2. **Transistor Sizing**: Optimize the size of transistors to achieve a balance between performance and power.
3. **Gate Minimization**: Minimize the number of logic gates used in the compressor to save power.
4. **Use of Asynchronous Logic**: This can help reduce power by avoiding unnecessary clocking.

b. Error Correction Implementation

- Integrate Hamming code logic or parity checks into the 4-2 compressor's output stage.
- Design the logic to handle error detection and correction seamlessly, ideally without significantly impacting performance.

3.7 Performance Analysis

- **Delay**: Consider the delay introduced by the error correction circuitry. Balancing error correction with speed is crucial.
- **Power Consumption**: Measure the power consumption of the multiplier, with and without the error correction, to evaluate trade-offs.

3.8 Summary

The 4-2 compressor is a crucial element in designing a low-power Dadda multiplier, significantly reducing the complexity of the summation process in binary multiplication. By carefully considering the implementation and optimization

techniques, designers can achieve a low-power, high-performance multiplier suitable for various applications. A low-power Dadda multiplier utilizing a 4-2 compressor with an integrated error-correcting module can significantly enhance the reliability of multiplication operations. By carefully implementing error detection and correction strategies, combined with low-power design techniques, it is possible to achieve both efficiency and accuracy in digital multiplication circuits. This design is particularly useful in applications where power efficiency and data integrity are paramount, such as in embedded systems and battery-operated devices.

In this chapter a comprehensive exploration of the design, methodology, and optimization strategies for developing a low-power, high-efficiency 8-bit Dadda multiplier using 4:2 inexact compressors—tailored specifically for image processing applications. The chapter lays the groundwork for efficient digital signal processing (DSP) systems by focusing on a balance between accuracy, area, power, and delay.

CHAPTER 4
Proposed Design and Model

4.1 Introduction

In this section, we present an idea of the proposed design and model for the development of a 4:2 inexact compressor-based 8-bit Dadda multiplier optimized for low power consumption and high performance. The design aims to address the increasing demand for energy-efficient computational units in image processing applications while maintaining acceptable levels of accuracy and throughput. Here's an outline of the key aspects covered in this section:

By introducing the proposed design and model in this section, we set the stage for a comprehensive exploration of the design methodology, implementation details, and performance evaluation of the 4:2 inexact compressor-based 8-bit Dadda multiplier optimized for low power consumption and application in image processing tasks [20].

4.2 Design and Architecture

In this section, we delve into the architectural design of the proposed 4:2 inexact compressor-based 8-bit Dadda multiplier optimized for low power consumption and tailored for image processing applications [44]. The architecture work flow serves as the backbone of the multiplier, dictating its structure, functionality, and performance characteristics. Here's a breakdown of the key components covered in this section:

1. A 8 bit approximate multiplier usind Dadda Multiplier algorithm is designed.

2. Approximate Compressors are used to reduce the Stages of the Multiplier [67].

3. Error correction module is used to reduce the error.

4. Output of the Adder is Applied to image processing .

5.Comparison of the proposed work with existing Literature.

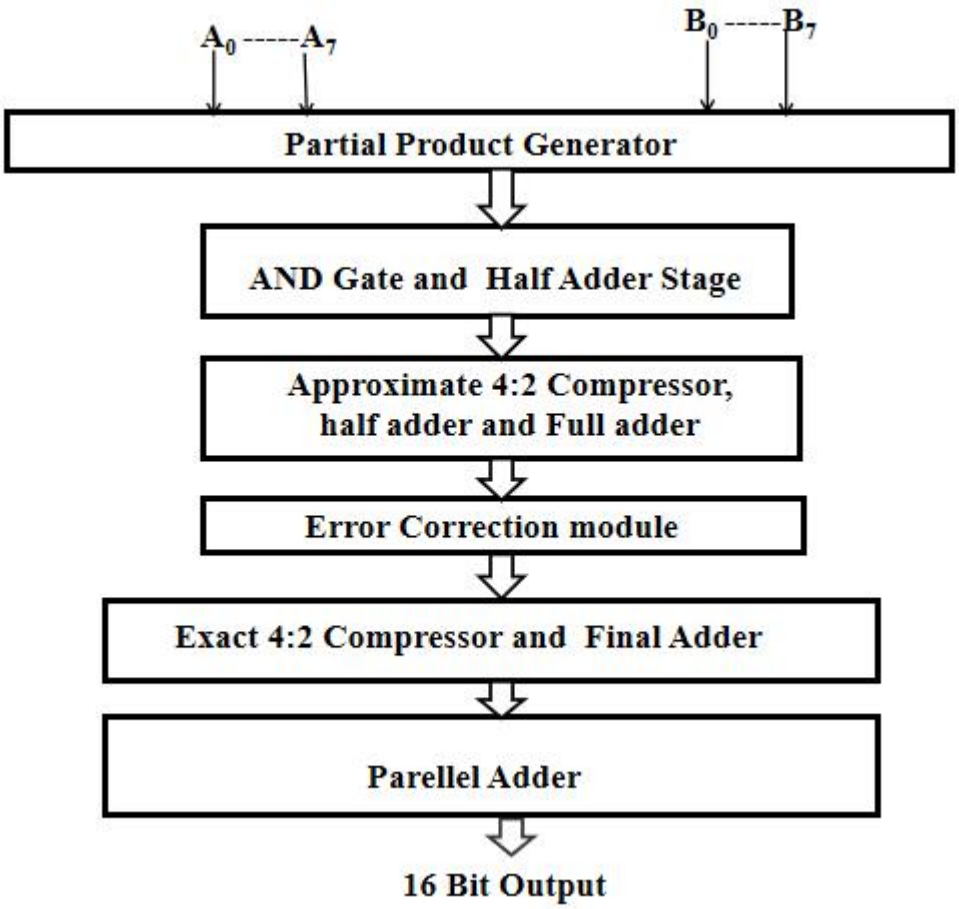


Figure 4.1 Dadda Multiplier Flow diagram with Compressor

A digital image application that is implemented using Verilog HDL. First of all, a jpg Image is taken as an input and then this image is converted into a binary image using Matlab and programming [67]. An image is converted into a gray-level image which is converted into a binary image (which is large). This large-sized image is resized and written into a text file. Simulation is checked out concerning the design summary and timing analysis. Gradients with magnitude are calculated in Verilog HDL synthesis for simulation. Now model sim read the text file which is generated using MATLAB in RAM, and then extracts the image window [68].

The proposed 4:2 compressor architectures are utilized in constructing an 8 × 8 Dadda multiplier using three compressors. The analysis aims to ascertain the effectiveness of the proposed designs, focusing on both accuracy and implementation efficiency

metrics. Our evaluation centers on assessing the power consumption metrics of the low-power multiplier, specifically the 4:2 compressor.

4.3 Tools and Technologies Utilized

During the research and achieving the objectives for the research proposed work, various tools and technologies were utilized. Xilinx ISE and Cadence 45nm CMOS technology tools were utilized for the design and implementation of the inexact multiplier [102]. These tools offer comprehensive development environments and synthesis. Verilog Programming Language: Verilog was utilized for the implementation of multiplier design. Verilog allows for the description and simulation of digital circuits, making it suitable for FPGA design and verification.

1. **Xilinx and Cadence 45nm:**

Xilinx ISE and Cadence 45nm CMOS technology tools were utilized for the design and implementation of the inexact multiplier [20]. These tools offer comprehensive development environments and synthesis.

2. **Verilog Programming Language:**

Verilog was utilized for the implementation of the multiplier design. Verilog allows for the description and simulation of digital circuits, making it suitable for FPGA design and verification.

3. **Image Processing Software:**

Image processing software tools Mat-lab were utilized for testing and validation of multiplier design in image applications. These tools facilitated the assessment of the multiplier's performance and efficiency in real-world image processing tasks.

Synthesis and optimization tools were employed to optimize the multiplier design for power efficiency and performance. These tools enable automated optimization of the design at both architectural and circuit levels. By leveraging these tools and technologies, the research was able to effectively address the objectives outlined for the proposed work, leading to the development of an efficient approximate low-power multiplier tailored for DSP processor applications in image processing [70].

4.4 The Proposed System

In this section, we propose the approximate 4:2 compressors UCAC1, UCAC2, and UCAC3 [69]. A high-probability input pattern is introduced, and the proposed designs are integrated into 8-bit multipliers based on the Dadda tree, as shown in Fig. 3

The Dadda multiplier offers several advantages over other multiplication algorithms [71]:

1. **Fast Operation**: By generating fewer partial products, the Dadda multiplier reduces the number of required additions, resulting in faster multiplication.
2. **Low Hardware Complexity**: Its regular and symmetric structure simplifies hardware design and reduces the number of logic gates needed.
3. **Scalability :** The Dadda multiplier can easily be scaled to support larger input sizes without significantly increasing hardware complexity.

However, the Dadda multiplier also has certain drawbacks:

1. **Higher area overhead**: Its regular and symmetric structure leads to a higher area overhead, especially for small input sizes.
1. **Higher Power Consumption**: The complexity of the required logic circuits results in higher power consumption compared to other algorithms.

The choice of multiplication algorithm depends on specific application requirements, such as input size, speed, and power constraints. Table 3 presents a comparative study of existing multipliers, including those based on carry look-ahead adders (CLA), Vedic mathematics, and Wallace trees. The CLA-based multiplier is the most area-demanding, followed by Vedic and Wallace tree multipliers [72].

In this work, we propose a digital image application implemented using Verilog HDL. Initially, a JPEG image is taken as input and converted into a binary image using MATLAB [73]. The image is first converted to grayscale and then to a binary format. This large binary image is resized and saved to a text file. The proposed work is implemented using Xilinx ISE and Modelsim. The simulation is evaluated based on design summary and timing analysis. Gradients with magnitude are calculated in Verilog HDL synthesis for simulation. Modelsim reads the text file

generated by MATLAB into RAM and then extracts the image window.

4.5 Approximate Compressor Based Multiplier

There are two types of compressors Exact and Inexact compressors. These compressors can be classified and designed in various bit sizes. In this work we are using 4:2 inexact compressor designs to validate and evaluate the 8 bit dadda low power approximate multiplier. In this work we three compressor designs named as UCAC1, UCAC2 and UCAC3 are implemented on dadda multiplier.

Table 4.1 displays the truth tables for the three compressors that have been suggested. Figure 4.2 shows the circuits of the proposed three rough compressors. C_{in}, C_{out} and Carry are all canceled in this work. The following is a list of UCAC1's logical purpose:

Table 4.1 Truth table for the proposed compressors.

A_1	A_2	A_3	A_4	UCAC1	UCAC2	UCAC3
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	1	0	1
0	1	0	0	0	0	1
0	1	0	1	1	1	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	0	1	1	1	1
1	0	1	0	1	1	0
1	0	1	1	1	1	1
1	1	0	0	1	0	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

Figure 4.2 shows the design of the compressors. Finally UCAC3 compressor design is the one we are considered for the implementation on the multiplier using cadence tool by which parameter of multiplier delay is reduced [69]. This delay efficient architecture of compressor design consist one OR gate as shown in figure 4.2.

$$Sum = y_1y_2 + (y_1 + y_2)(y_3 + y_4) + y_3y_4 . \quad (4.1)$$

The approximate 4:2 compressors UCAC1, UCAC2 and UCAC3 are used in dadda multiplier. A big probability input pattern is then supplied.

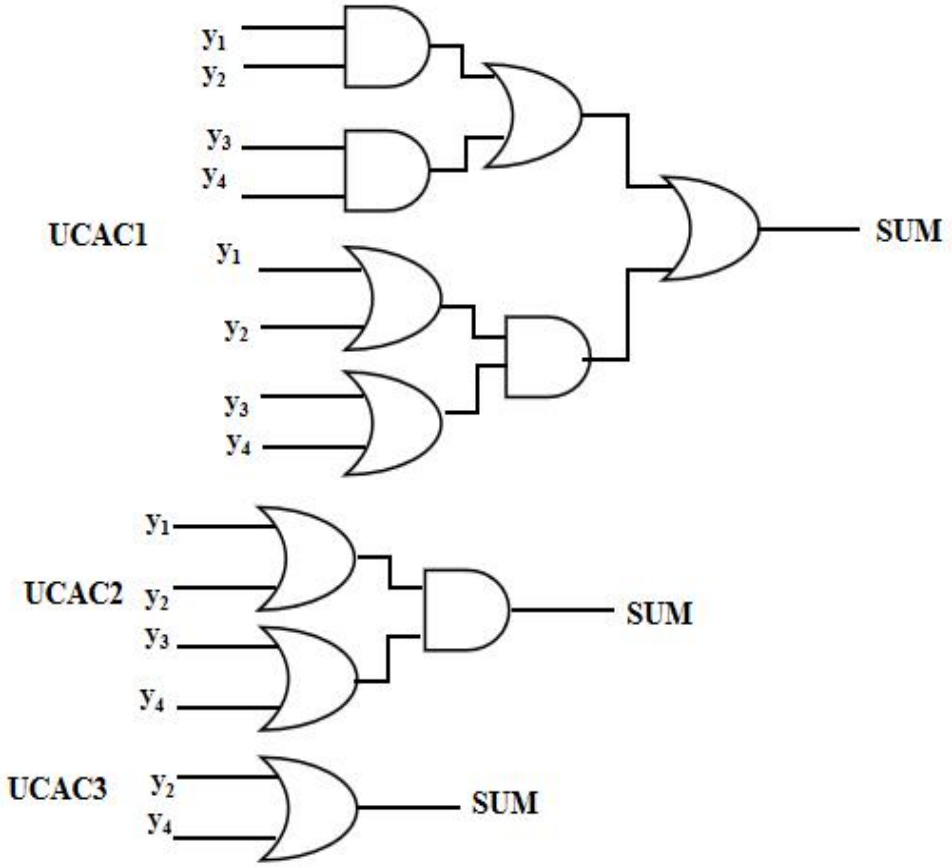


Figure 4.2: Basic diagrams of the implemented 4-2 compressors.

Figure 4.2 shows the circuits of the proposed three rough compressors. C_{in}, C_{out} and Carry are all canceled in this work. The following is a list of UCAC1's logical purpose: Figure 4.2 shows the design of the compressors. Finally UCAC3 compressor design is the one we are considered for the implementation on the multiplier using cadence tool by which parameter of multiplier delay is reduced. This delay efficient architecture of compressor design consist one OR gate as shown in figure 4.2.

4.6 Error-correcting module (ECM)

To create a positive error when the suggested compressors are integrated into the multipliers, a constant logic “1” is added to the following bit of the CC’s MSB. The 0 scenario has a nonzero probability of 81/256.

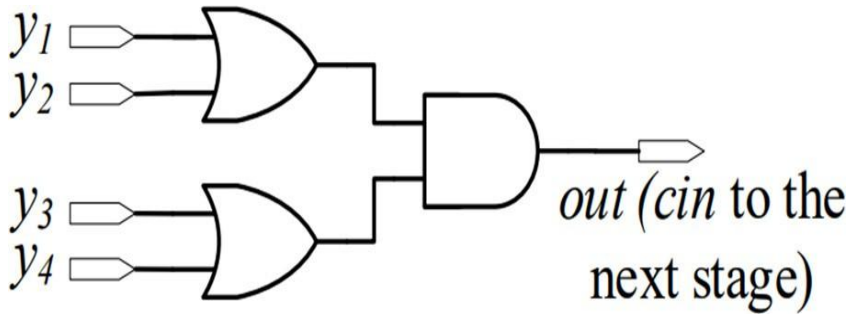


Figure. 4.3. The circuits of the error-correcting module [69].

However, in the 0° scenario, the design produces the correct result. In this case, there is no negative error to counteract the positive error introduced by the correcting bit. Additionally, as shown in Figure 4.3, the correcting bit contributes significantly to the total Error Distance (ED) due to its relatively high binary weight. Consequently [69]. an Error Correction Mechanism (ECM) is proposed to identify this input pattern and change the correcting bit from logic "1" to logic "0" in this specific instance. While the compressors cover a broader range of scenarios, they introduce a negative error that can counterbalance the non-zero correcting bit. Figure 4.3 illustrates the ECM circuit. The inputs to the ECM are identical to those of the approximation compressor, including the Most Significant Bit (MSB) of the Carry Chain (CC). When the 0° scenario is detected, the circuit outputs logic "0".

Although approximation compressors can replace all classic 4:2 compressors, their error performance suffers. Therefore, in the proposed multipliers,$N-1$ less significant bits utilize the approximation compressors.

A synthesized net listed diagram for 8*8 approximate multiplier is show in figure 4.4. The results of this phase affirmed the functionality and accuracy of the inexact compressor design within the specified technology parameters

By employing these parameter optimization strategies, we aim to design a low-power approximate multiplier architecture that strikes an optimal balance between area efficiency, timing performance, and power consumption, thereby fulfilling the objectives outlined for the proposed design and model [100].

4.7 Design of the Proposed Architecture of Dadda Multiplier

The approximate 4:2 compressors UCAC1, UCAC2 and UCAC3 are used in dadda multiplier. A big probability input pattern is then supplied [69].

The suggested designs of Approximate Compressors are also included in 8-bit multipliers .8-bit multipliers (N = 8) are created to evaluate these blocks following the proposed approximation compressors and ECM, which are intended to streamline and speed up the compression process [69].

(1)MUL1: UCAC1 multiplier and ECM;

(2)MUL2: UCAC2 multiplier and ECM;

(3)MUL3: UCAC3 multiplier and ECM.

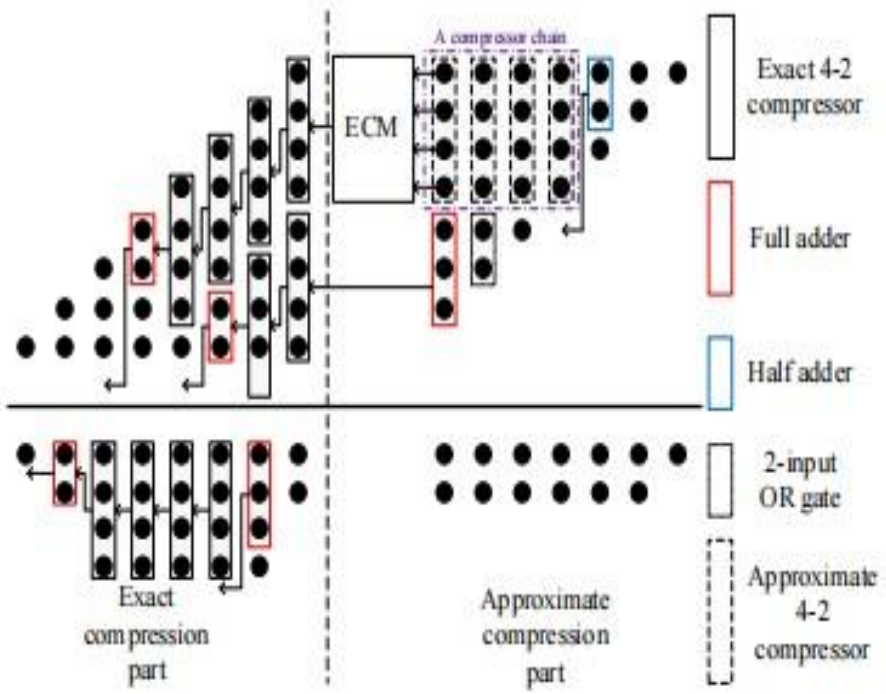


Figure. 4.4 Dadda multiplier using the proposed designs[69]

We implement three compressor designs—UCAC1, UCAC2, and UCAC3—on the Dadda multiplier. The figure 4.4 illustrates the design of these compressor based low poer multiplier. Among these, the UCAC3 compressor design is selected for implementation on the multiplier using the Cadence tool, which significantly reduces the multiplier's delay. This delay-efficient. Multiplier design is proposed the reduction in delay using cadence 45nm to get the desired result. Which is compared with the literature for the novelty of the work.

We implemented Dadda UCAC2 multiplier at slow and fast corner. The number of cells employed in the design is pretty consistent throughout the fast and slow process types, with 278 cells for the fast process and 292 cells for the slow process [68]. Leakage power is the power consumed by a circuit when it is not actively switching [69]. The leakage power in the rapid process is significantly larger at 130.024 nw than in the slow process, which has a much lower leakage power of 36.988 nw. This disparity could be related to differences in transistor properties and leakage currents between the manufacturing versions. Internal power refers to the power dissipated as a result of switching activity within the circuit.

The fast process consumes more internal power (590267.421 nw) than the slow process (348197.706 nw). This mismatch may be due to changes in gate delays and propagation times between the process types. Net power is the total power dissipation in the circuit, including both leakage and internal power components. The net power consumption of the rapid process is significantly larger, at 200555.701 nw, than that of the slow process, which is 104499.728 nw. Total power is the sum of leakage power, internal power, and any additional power components in the circuit. The rapid process consumes much more overall power, 790823.122 nw, than the slow process, which consumes 452697.434 nw . The fast process generally consuming more power across all metrics compared to the slow process.

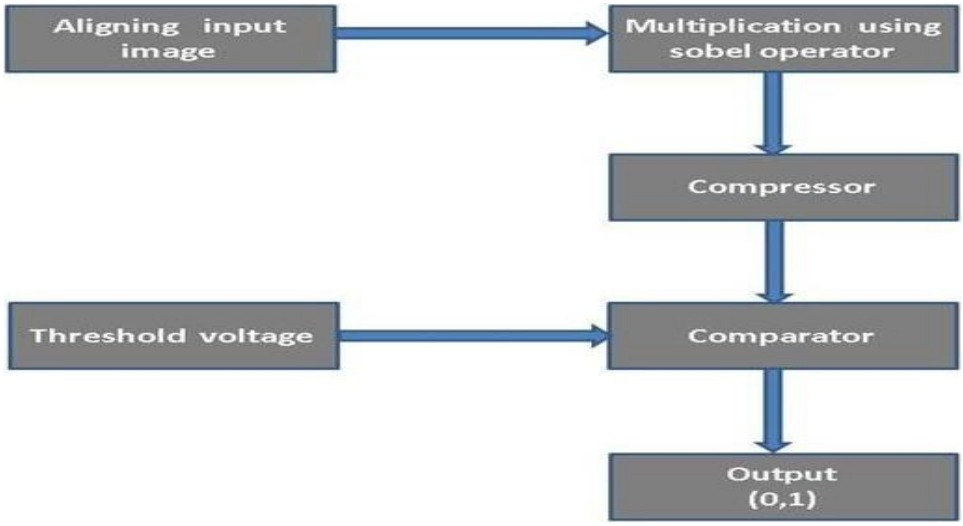


Figure. 4.5 Sobel edge detection block diagram

Converting an RGB image to a binary image involves several steps, typically including loading the image, converting the RGB values to grayscale, and then thresholding the grayscale image to create a binary representation.After converting the RGB image to binary, you can apply a Dadda multiplier to perform operations like image filtering, edge detection, or other transformations that require multiplying pixel values. The Dadda multiplier is advantageous in this context due to its efficiency and low power consumption. Figure shows the Procees of Image Conversion.

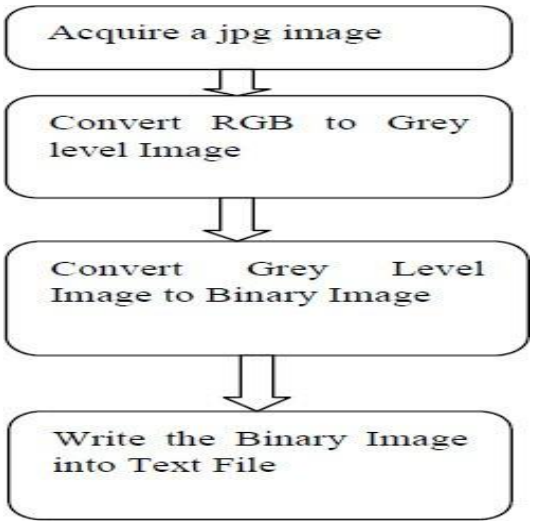


Figure. 4.6 Process flow of RGB to binary image conversion (MATLAB Part)

The suggested compressors,however, produce the right result in the 0° scenario. In

this scenario, there is non- negative error to cancel out the positive error caused by the correcting bit. In addition, concerning Fig. 7, the correcting bit has a substantial total ED due to its relatively high binary bit weight. As a result, an ECM is offered to recognize this input pattern and change the corrective bit in this instance from logic "1" to logic "0". Although they cover a greater likelihood in other circumstances, the compressors produce a negative error that can cancel out the nonzero corrective bit. In Fig. 4, the ECM's circuit is depicted [69]. The inputs to the ECM are identical to those of the approximation compressor, the MSB of the CC. When the 0° scenario occurs in its inputs, this circuit merely outputs logic "0".

Although approximation compressors can replace all classic 4:2 compressors, the error performance suffers.[5] As a result, $N - 1$ less significant bits in the suggested multipliers use the approximation compressors.

4.8 Simulation Results and Discussion for parameter optimization

Among the various designs, Dadda UCAC3 demonstrates the best performance. It remains compact across all multipliers and significantly enhances the power performance of approximate multipliers compared to other designs. Verilog supports a function to read image data.

Table 4.2 Performance Metrics of Dadda Multiplier

	Cell Count	**Delay**			**Power**
	Gates	**Overall Delay ns**	**Gate Delay Ns**	**Path Delay ns**	**Power mw**
Dadda	834	31.984	14.17	17.80	239
Dadda Conventional	852	32.264	14.17	18.08	239
DADDA UCAC1	852	31.479	14.17	17.30	256
DADDA UCAC2	852	30.889	14.17	16.71	148
DADDA UCAC3	882	32.628	14.73	17.89	129

The synthesis process for the aforementioned multiplier involves several critical steps aimed at transforming the high-level design description into a physical

implementation ready for fabrication. Table 4.2 shows the performance of the proposed design using xilinx ISE using compressor.

The image data are stored in a binary text file, which is then read and saved into memory as RGB image data for processing at a 50 MHz clock frequency.

In this section, the four suggested multipliers undergo evaluation for image multiplication. MATLAB software is utilized to dissect images into their components, which are then processed using approximate multipliers for multiplication. Subsequently, MATLAB amalgamates the data from the simulation tools to generate a new image. To gauge the effectiveness of the multipliers, the Peak Signal-to-Noise Ratio (PSNR) is employed. The edges of the output images generated by the proposed multiplier. Additionally showcases the design parameters utilized for the image application.

In the realm of architecture, the Dadda multiplier is susceptible to errors that can undermine its accuracy and reliability. The error metric of a Dadda multiplier denotes the degree of accuracy in its output compared to the expected result. This metric holds significance as it aids in evaluating the performance of the proposed multiplier and pinpointing any potential sources of error affecting its output. To assess the error performance of the Dadda multiplier, various metrics are available, including Mean Absolute Error (MAE), Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). These metrics serve the purpose of comparing the Dadda multiplier's output with the ideal output.The error detection outcomes are presented in Table 4.3.

The noise reduction efficacy of the algorithm is assessed using PSNR, MSE, and SSIM metrics across three noisy images with sigma values of 20, 30, and 40 [102]. The PSNR gradually decreases with higher noise intensity, suggesting that the non-local mean filtering algorithm is more suited for low-noise environments. For sigma = 20, the algorithm effectively eliminates noise while preserving image details and structure. This study underscores the pivotal applications of the non-local mean filtering algorithm in various industries, including the medical field.

Table 4.3 Error detection with various parameters.

Sigma	Ref [102]			Proposed Work		
	PSNR (dB)	MSE	SSIM	PSNR (dB)	MSE	SSIM
20	25.47	184.39	0.46	22.01	148.76	0.76
30	22.01	408.78	0.33	20.58	376.97	0.53
40	19.64	705.94	0.25	16.54	564.78	0.48

Furthermore, the results demonstrate that the Dadda multiplier exhibits a higher error rate compared to the Booth multiplier, albeit lower than the Wallace multiplier [102]. The integration of an Error Correcting Module (ECM) further enhances error metrics to achieve precise results. Notably, the ECM of the Dadda multiplier incurs lower overhead compared to other error correction mechanisms.

4.9 Summary

This chapter provides a comprehensive design, development, and performance analysis of a 4:2 inexact compressor-based 8-bit Dadda multiplier, focusing on low power consumption, high performance, and applicability in image processing.The chapter successfully demonstrates that the UCAC3-based Dadda multiplier offers a power-efficient, delay-optimized, and application-ready hardware solution for image processing tasks. The integration of the Error Correction Module (ECM) significantly improves error resilience, and the simulation outcomes confirm the effectiveness of the proposed architecture in real-world image processing scenarios.The chapter concludes that UCAC3-based Dadda multiplier achieves a superior balance between delay, power, and accuracy, making it highly suitable for low-power image processing applications in devices like embedded vision systems or portable imaging devices.This establishes the groundwork for efficient inexact computing architectures in digital signal processing (DSP), particularly where energy is constrained but a small error margin is acceptable.

CHAPTER 5

Experimental Results

5.1 Introduction

The investigation and Evaluation of compressors and multipliers utilize for the low power applications and implemented using the Verilog programming on the tool Xilinx and Cadence 45nm technology has yielded compelling results. Results of the key findings and outcomes achieved to pursuit the objective for this research work is presented in this chapter.

This section continues into the Parameter achieved and performance of applied approximate multipliers and compressors. The recommended compressors designed utilizing Cadence (45-nm CMOS technology) using both slow and fast process corners, working at 100 MHz and 1V supply voltage of 1 V. The analysis includes area, delay, power, latency, and PDP. Area represents the extent to which the recommended layout optimizes hardware, thus improving compactness. An exceptional design aims to optimize factors such as area, power and delay within a comparable range of precision, ensuring efficiency over varied range of accuracy.

Operating Parameters for Categorization:

The establishment of operating parameters for categorizing approximate low-power multipliers has provided a systematic framework for evaluating and comparing different multiplier designs.

Algorithm Design:

A novel algorithm tailored for the low-power multiplier has been devised, leveraging the Verilog programming language and innovative approximation techniques. The algorithm demonstrates promising results in reducing power consumption while maintaining computational accuracy, laying the foundation for efficient computation in resource-constrained environments.

Power Optimization Techniques:

Meticulous optimization efforts at both architectural and circuit levels have led to significant improvements in power efficiency without compromising performance. Through the integration of advanced optimization techniques, as parallelism, pipe-

lining, and voltage scaling, notable reductions in power consumption have been achieved, enhancing the overall performance of the proposed Inexact multiplier.

Implementation and Application in Image compression and processing:

The successful implementation of the multiplier design using Xilinx and Cadence 45nm technology validates its practical feasibility.

In summary, the exploration of the proposed objectives has yielded promising results, laying the groundwork for the development of energy-efficient computing solutions. The following sections delve into the detailed analysis and discussion of the obtained results, providing insights into the performance and efficacy of the designed multiplier in achieving the outlined objectives.

5.2 Performance Analysis and Evaluation

The performance of the proposed 4:2 inexact compressor-based 8-bit Dadda multiplier is crucial in assessing its effectiveness in meeting the objectives outlined for the research. This evaluation focuses on the key parameters of area, delay, and power, which are vital for determining the efficiency and suitability of the multiplier for image processing applications.

1. **Area Efficiency:** The area efficiency of the multiplier is assessed by quantifying the amount of hardware resources it occupies on the FPGA.

2. **Delay: D**elay of the multiplier refers to the time taken for the output to be generated in response to input signals. This parameter is critical for real-time image processing applications, where low latency is essential.Simulation results are analyzed to determine the propagation delay.

3. Power Consumption: It is a key consideration for low-power applications, especially in battery-operated devices or embedded systems.Power analysis.Additionally, the performance evaluation provides valuable insights for further optimization and refinement of the multiplier architecture for image processing applications.

5.2.1 Performance Analysis Using Xilinx: Figure 5.1 shows the RTL view of the 4-bit multiplier, which is an array multiplier. The RTL view of the 8×8-bit Dadda multiplier using the 4:2 compressor is shown in Figure 5.2. Figures 5.3 and 5.4 show the output waveforms of the 8-bit Dadda multiplier for different input variables.

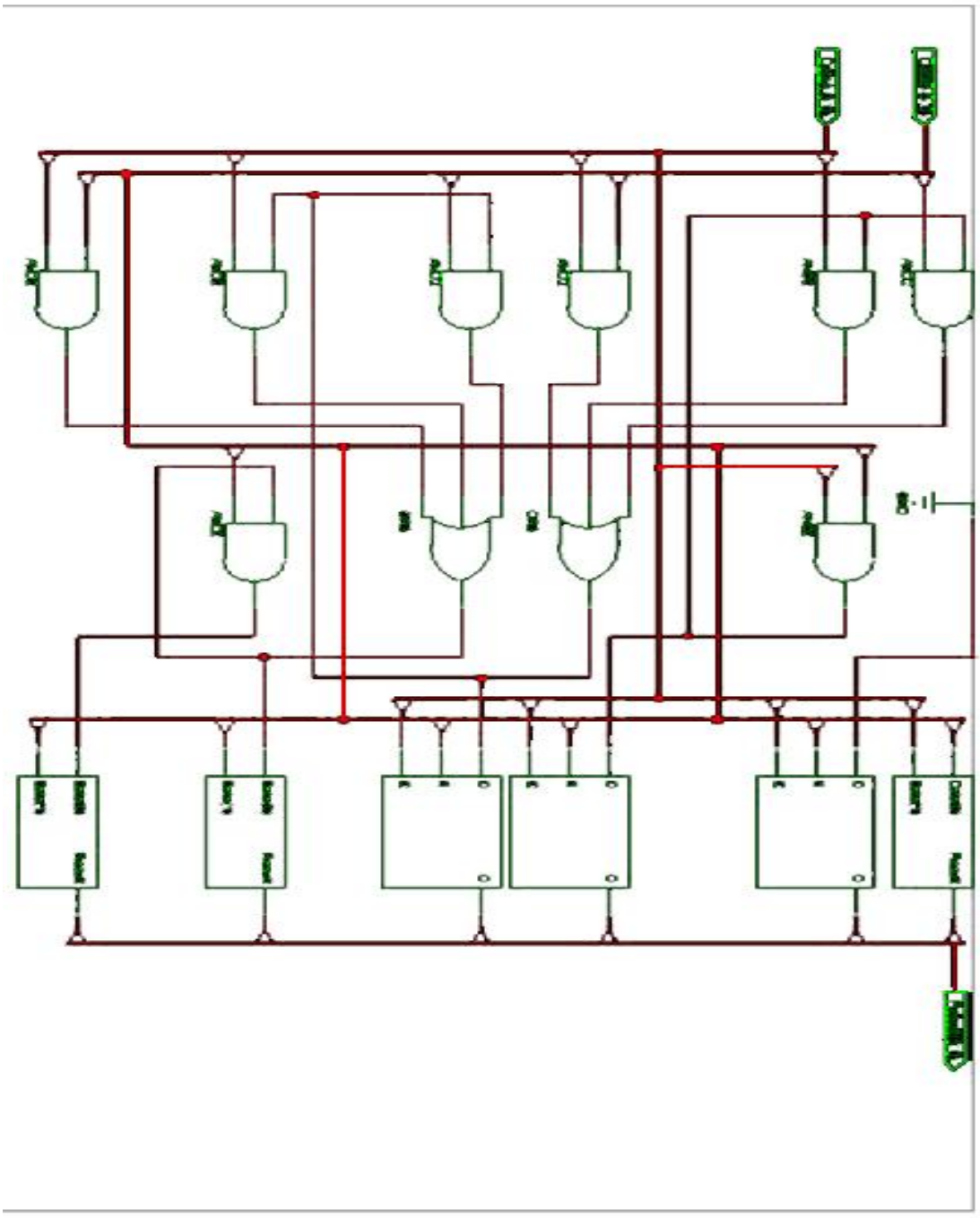


Figure 5.1 RTL View of 4 Bit Multiplier

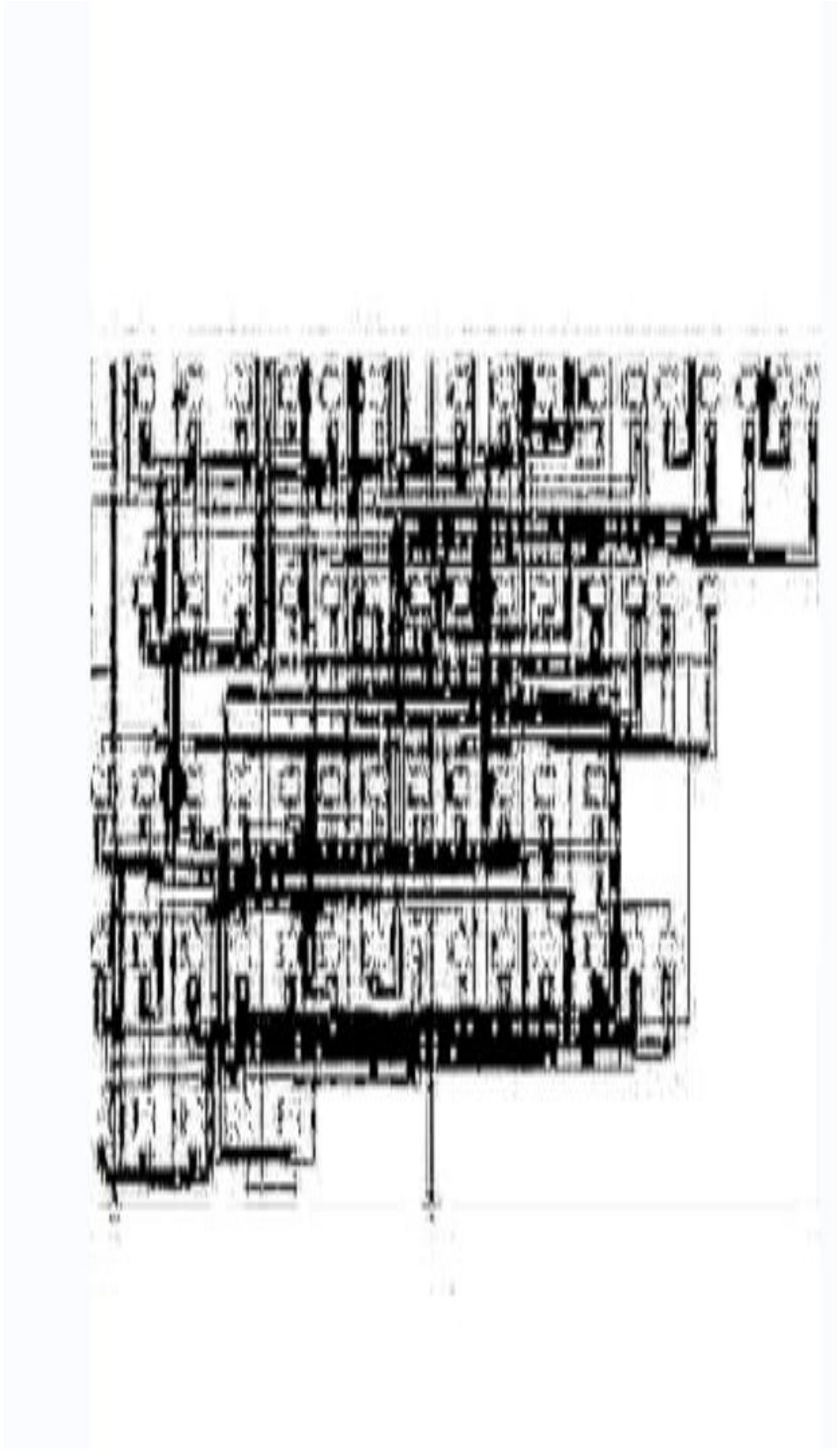


Figure 5.2 RTL VIEW OF 8 BIT MULTIPLIER

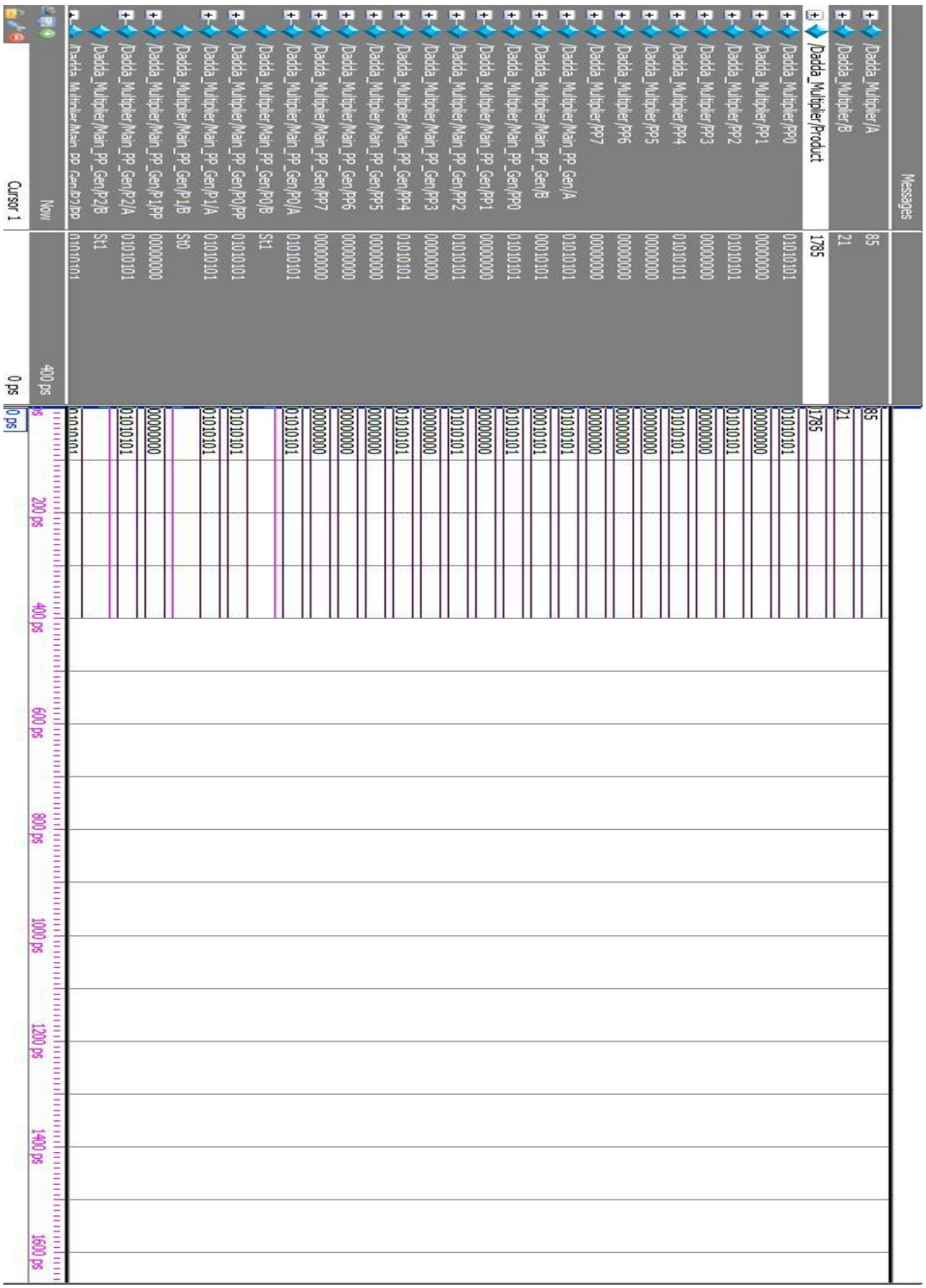


Figure 5.3 Waveform for 8 bit Dadda Multiplier

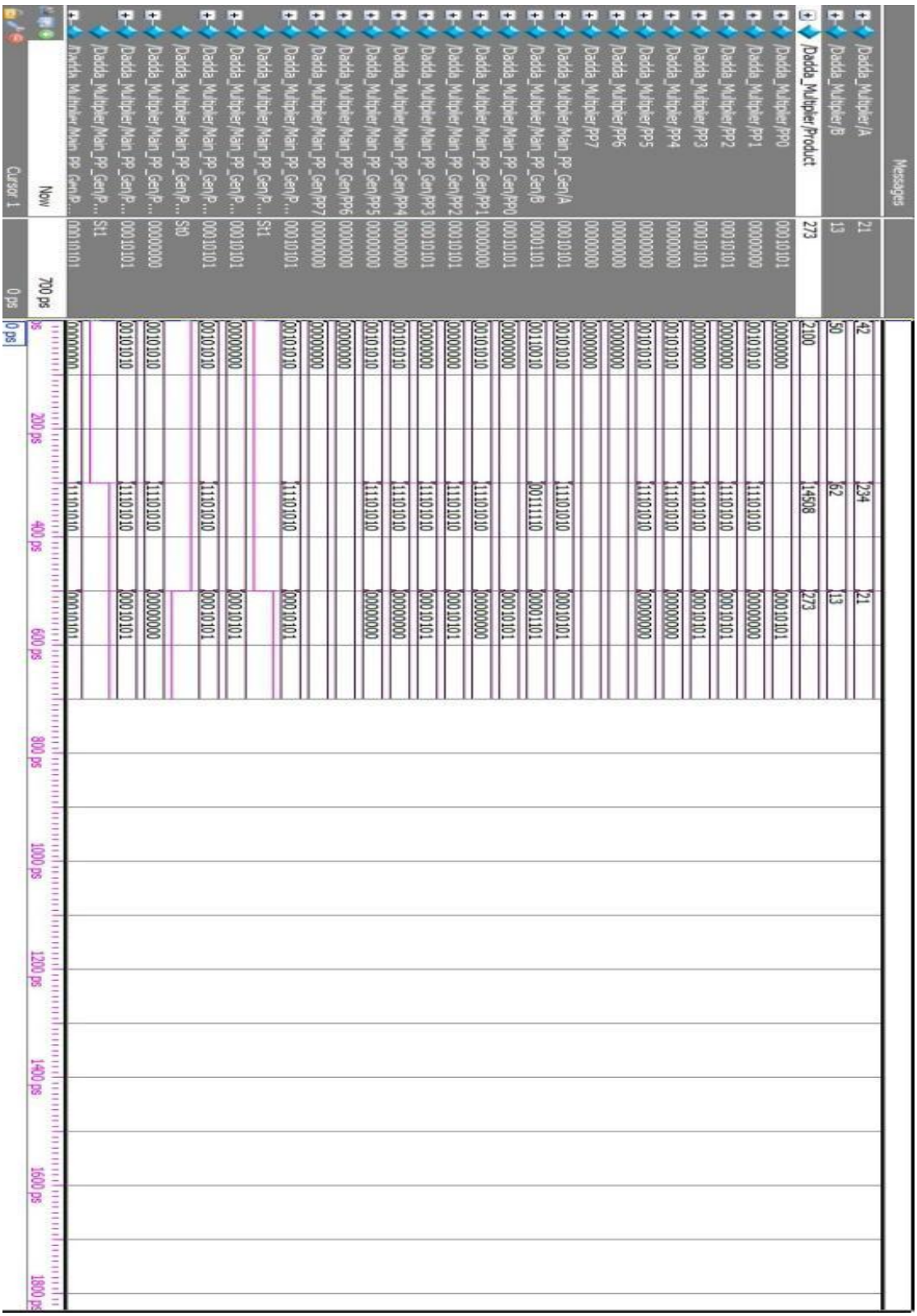


Figure 5.4 Waveform for 8 bit Dadda Multiplier with different Parameter

5.2.2 Performance Analysis Using Cadence 45nm for fast.lib

Figure 5.5 shows the RTL view of the compressor based 8 bit dadda multiplier using cadance 45nm for fast.lib. Simulation setup shown in the figure 5.6.

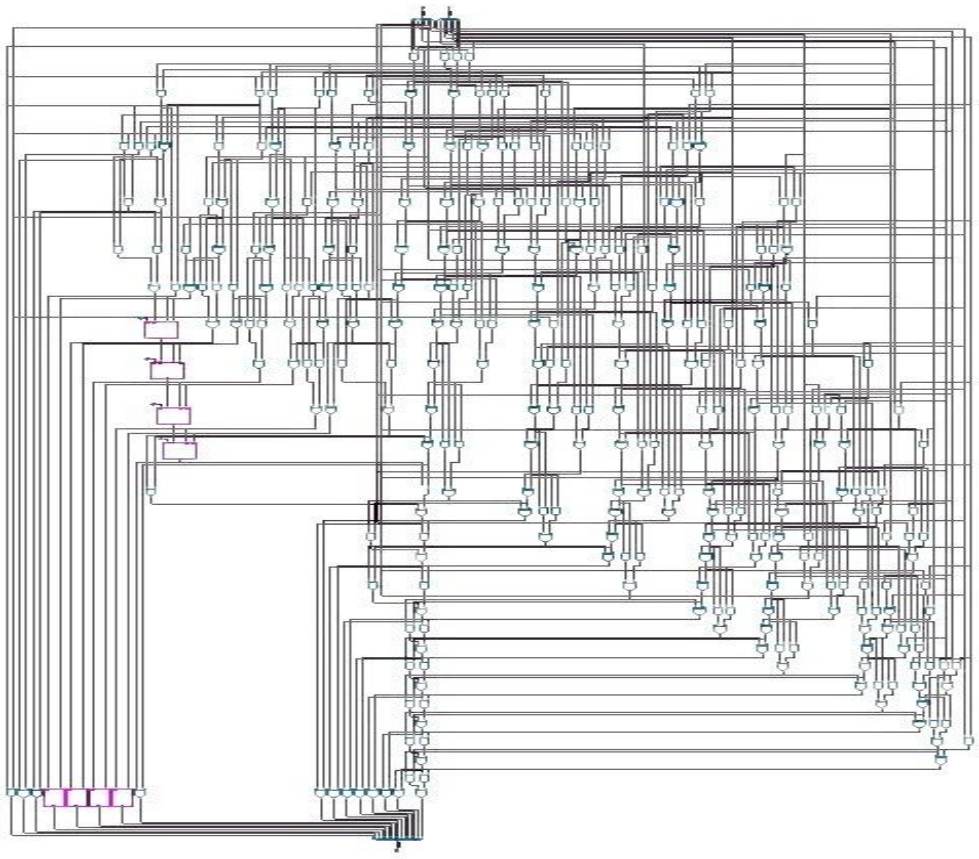


Figure 5.5 RTL View of the Compressor based 8 Bit Multiplier for fast.lib

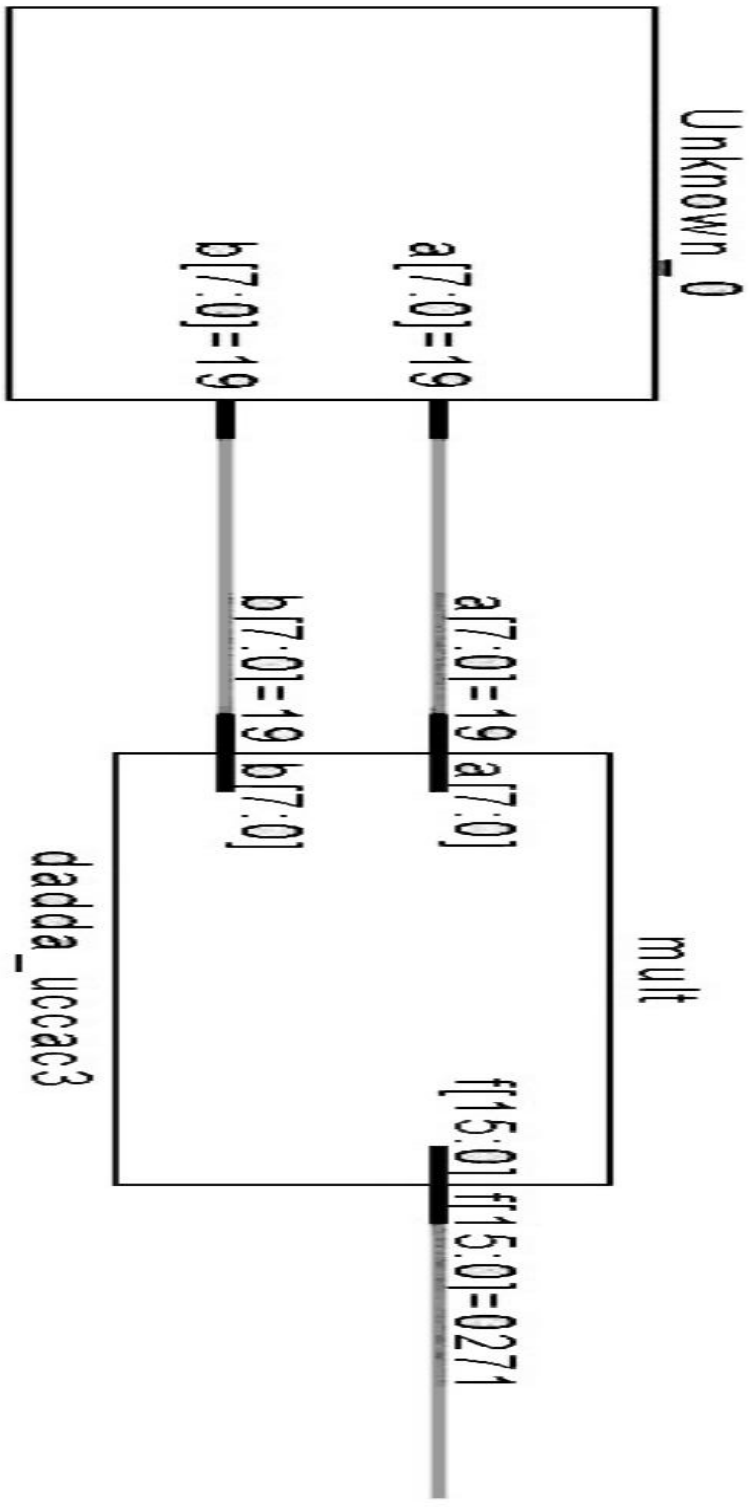


Figure 5.6 Simulation setup for 8*8 multiplier for fast.lib

5.2.3 Performance Analysis Using Cadence 45nm for slow.lib:

Figure 5.7 shows the RTL view of the compressor based 8 bit dadda multiplier using cadance 45nm for slow.lib. Output waveform shown in the figure 5.6 and in figure 5.7.

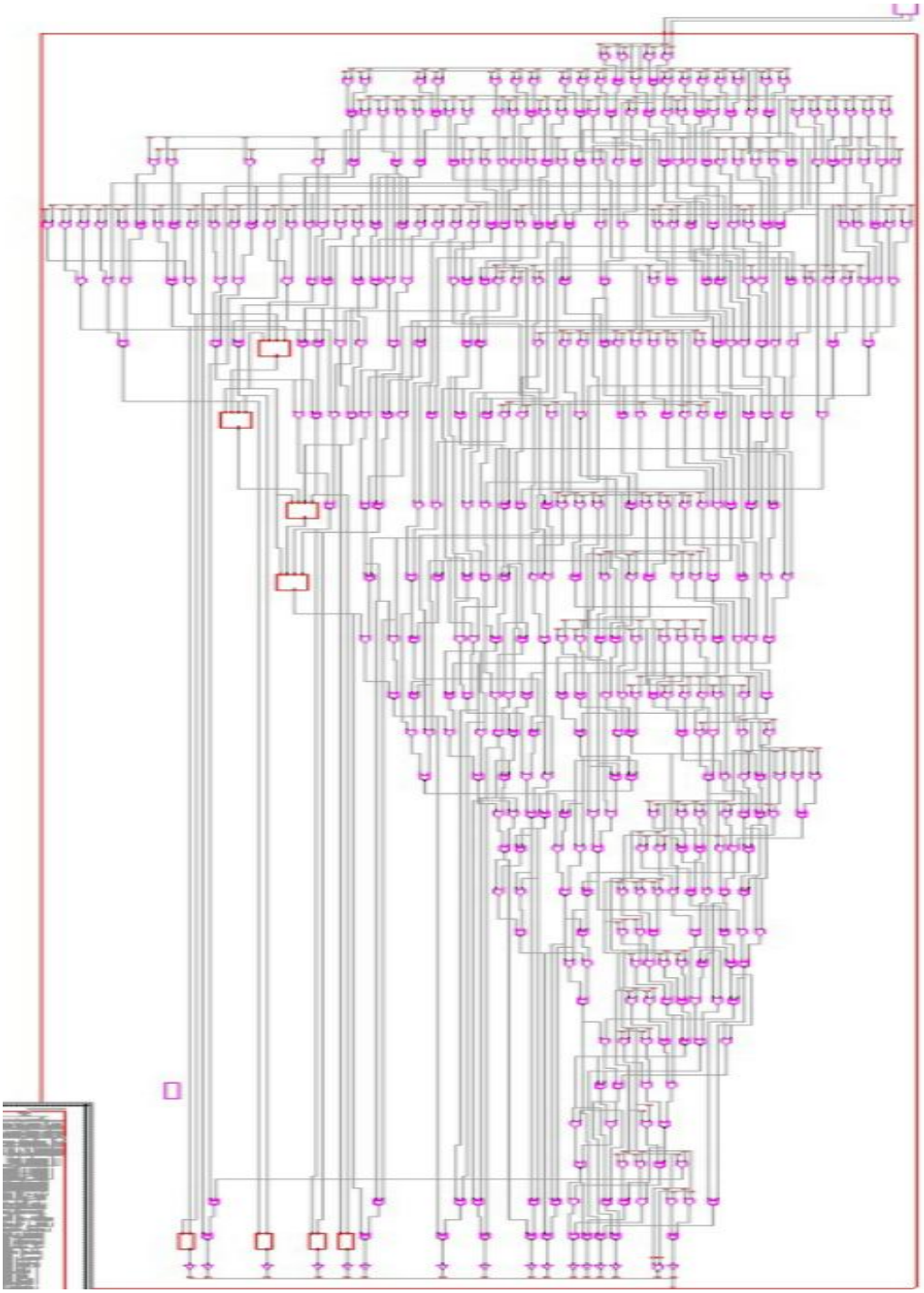


Figure 5.7. RTL View of the Compressor based 8 Bit Multiplier for slow.lib

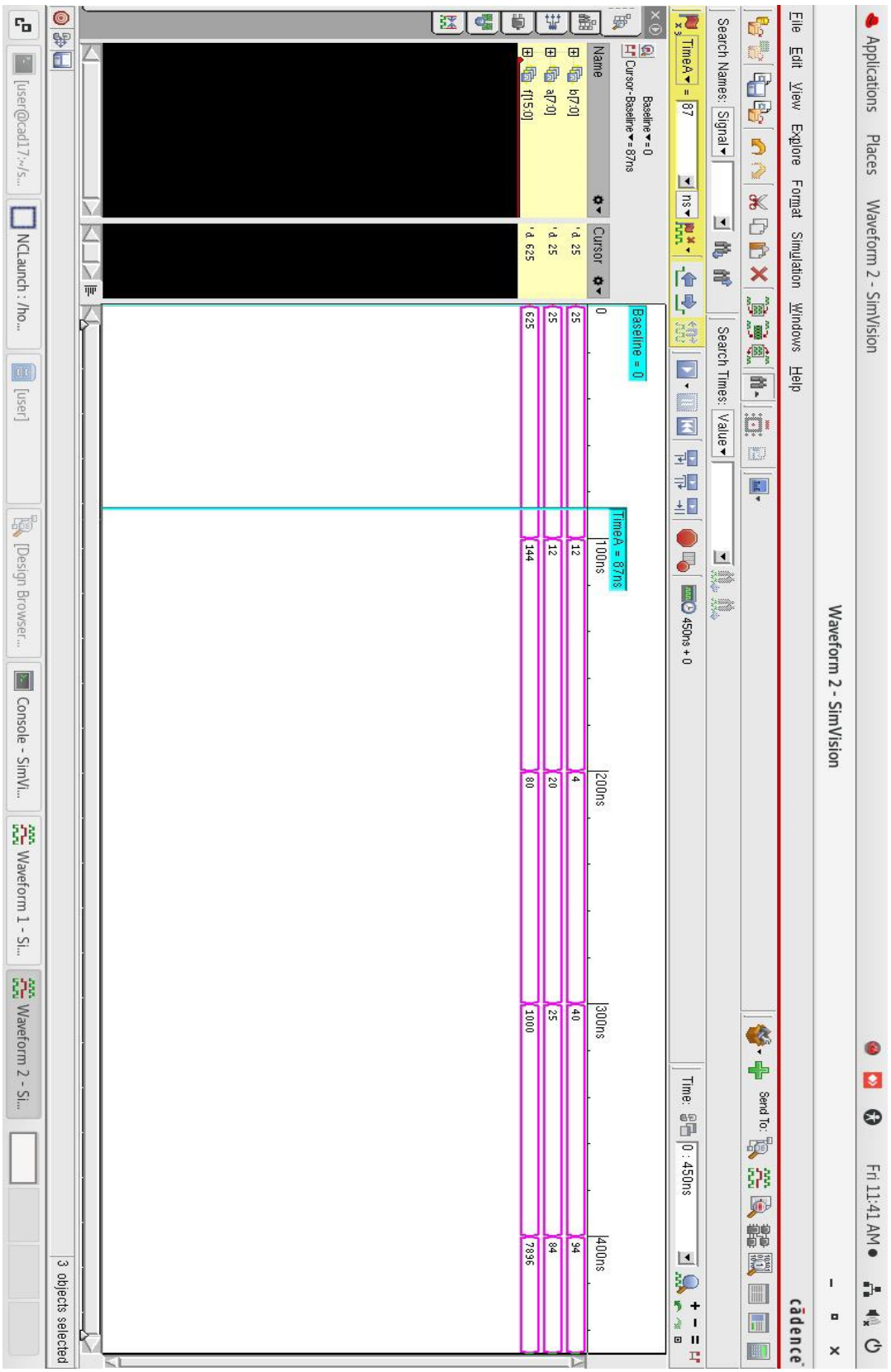


Figure 5.8 Out Put Waveform of 8 bit low power Multiplier for slow.lib

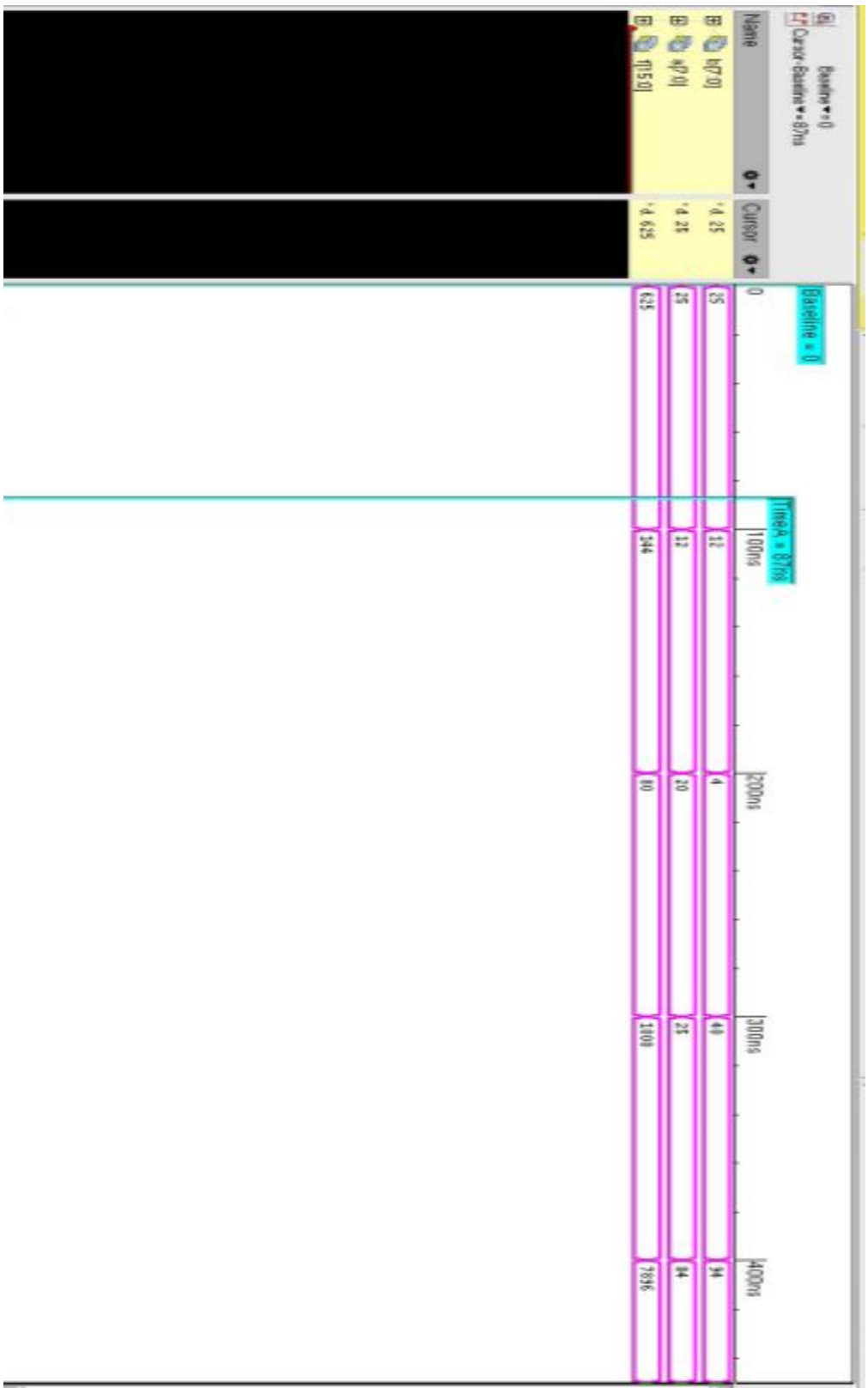


Figure 5.9. Out Put Waveform of 8 bit low power Multiplier with input and output

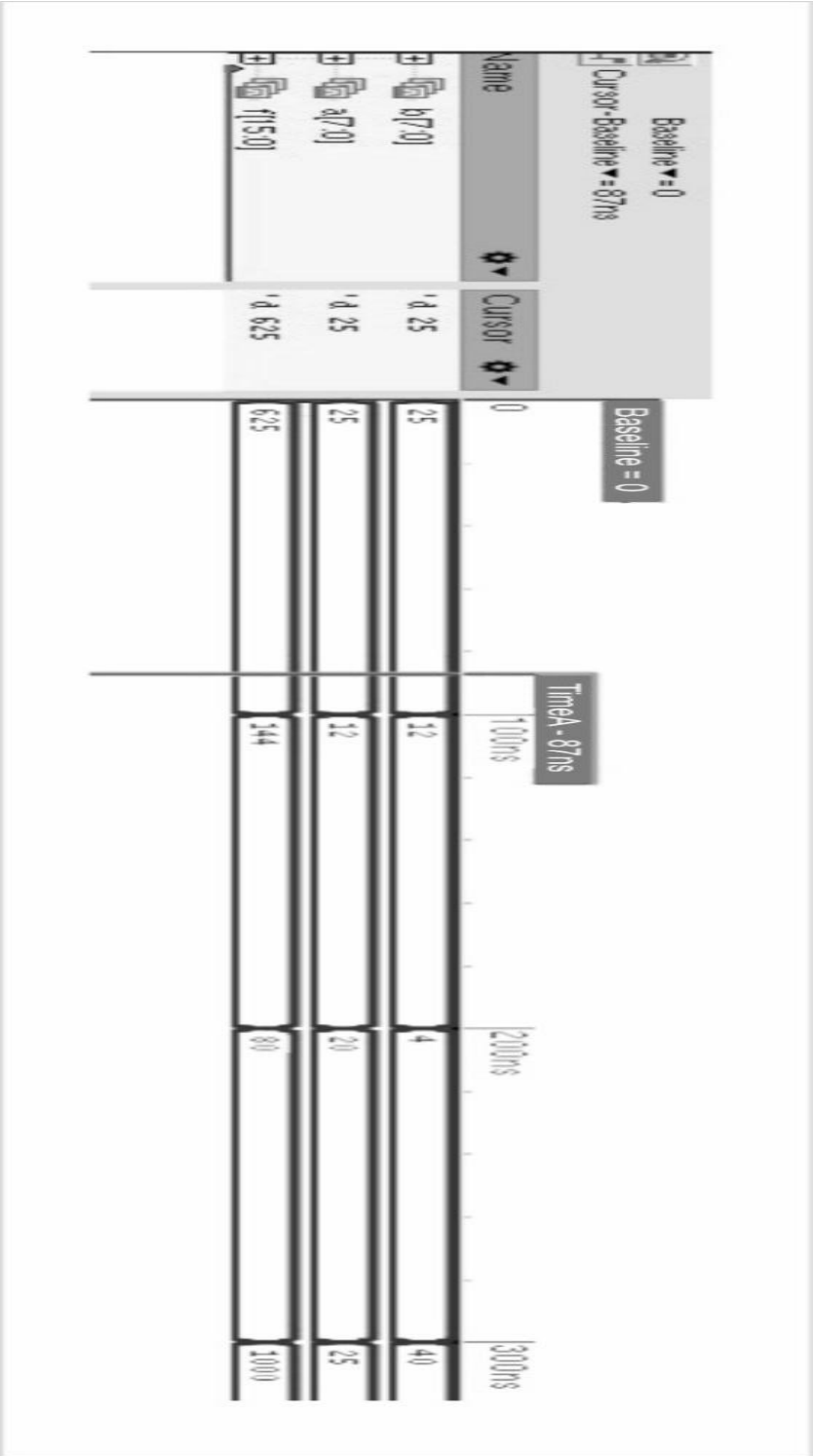


Figure 5.10. Close view Out Put of 8 bit low power Multiplier with Input and Output

Figure 5.10.shows the Close view Out Put of 8 bit low power Multiplier with Input and Output. In this we take different input variable to get the multiplication.

5.3 Parameter Result Using Cadance

5.3.1 Parameter results Using Cadence 45nm for fast.lib

Cell Report for fast.lib and Graphical Representation: Figure 5.11 shown the Cell report for the fast.lib for the compressor based dadda Multiplier. Figure 5.12 show the graphical representation of the cell result.

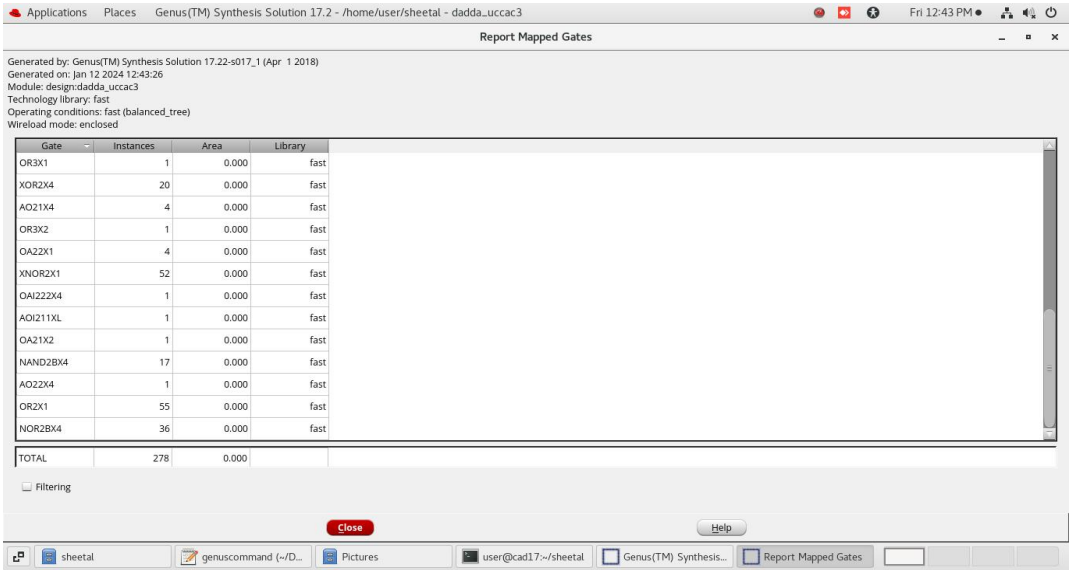


Figure 5.11 Cell Report for fast.lib

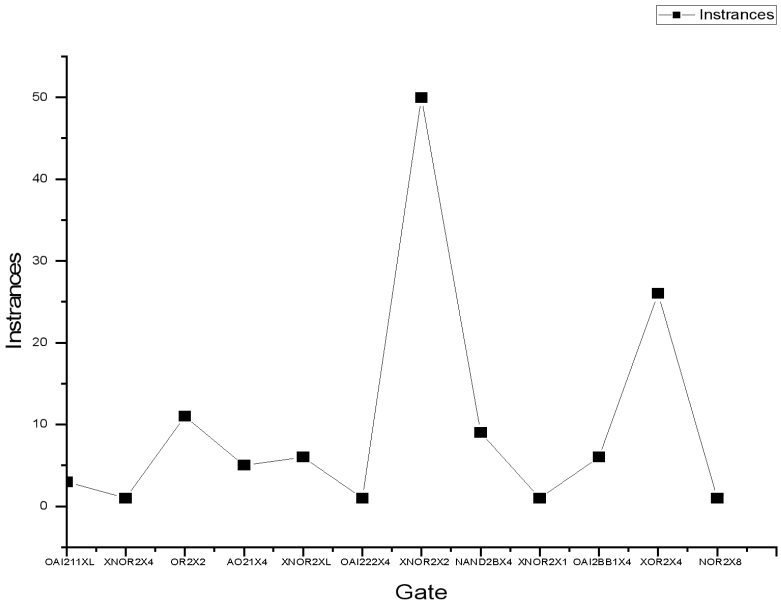


Figure 5.12 Graph for Cell Report for fast.lib

Power synthesis Report for fast.lib and graphical representation: Figure 5.13 shown the Power report for the fast.lib for the compressor based dadda Multiplier. Figure 5.14 show the graphical representation of the Power result.

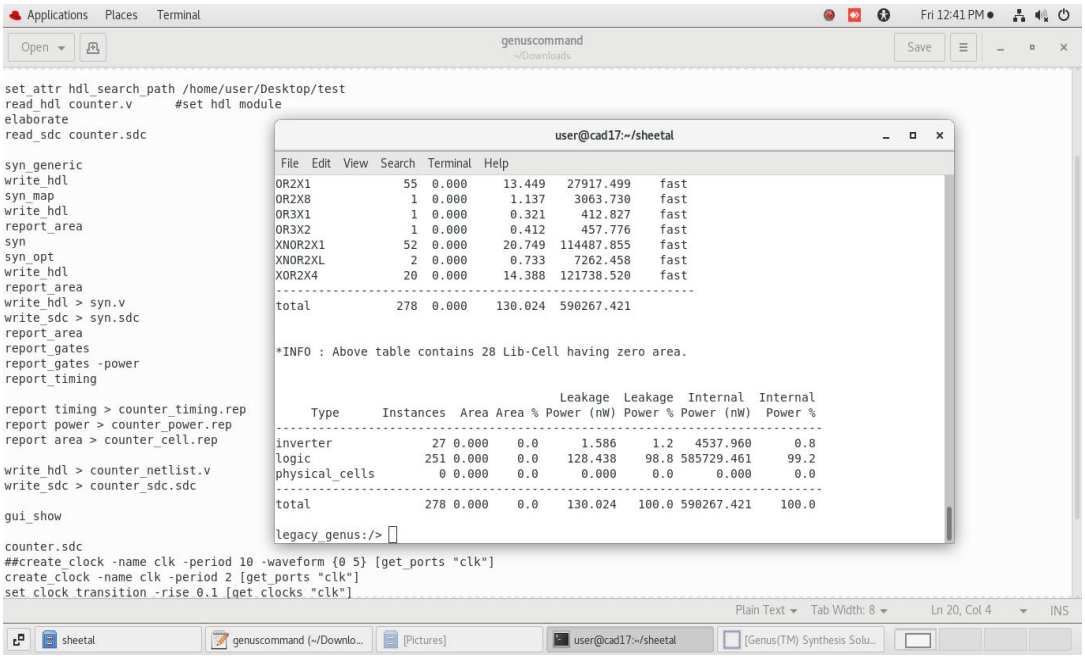


Figure 5.13 Power Synthesis Report for fast.lib

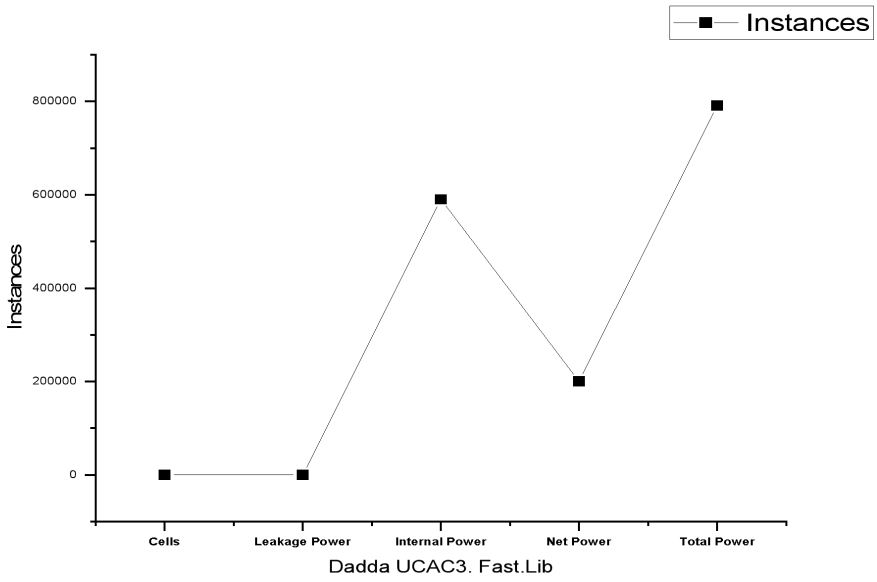


Figure 5.14 Graph for Power Synthesis Report for fast.lib

5.3.2 Parameter results Using Cadence 45nm for slow.lib

Synthesis Results for slow.lib: Timing Synthesis report for 8*8 Dadda Multiplier using 45 nm CMOS technology.

Cell Report for Slow.lib: Figure 5.15 shown the cell report for the slow.lib for the compressor based dadda Multiplier. Figure 5.16 show the graphical representation of the cell result.

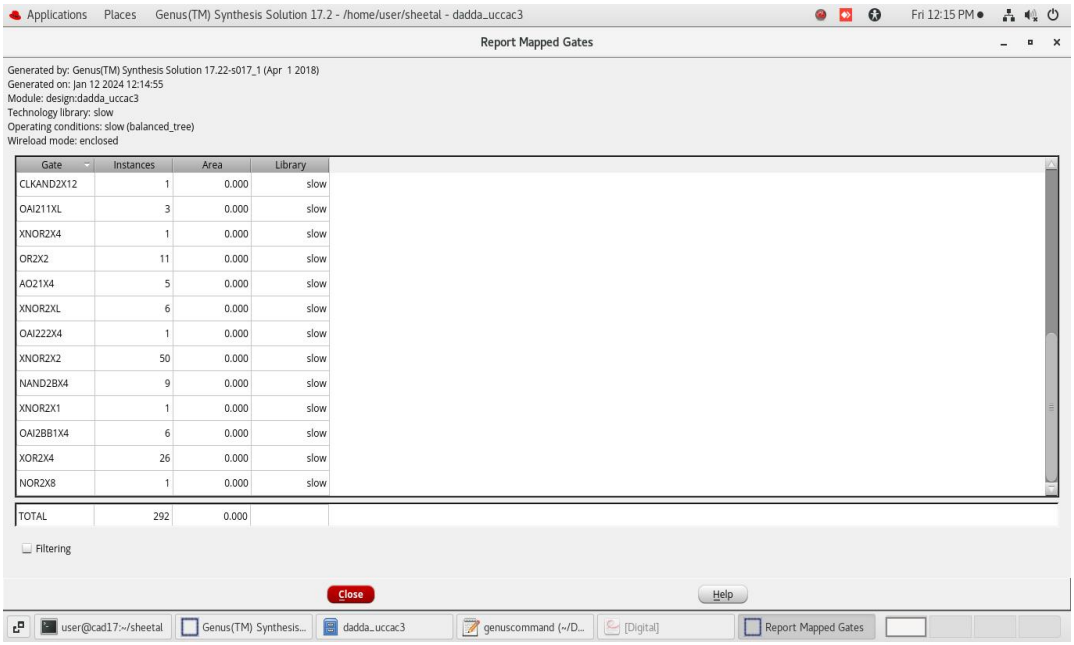


Figure 5.15 Cell Report for slow.lib

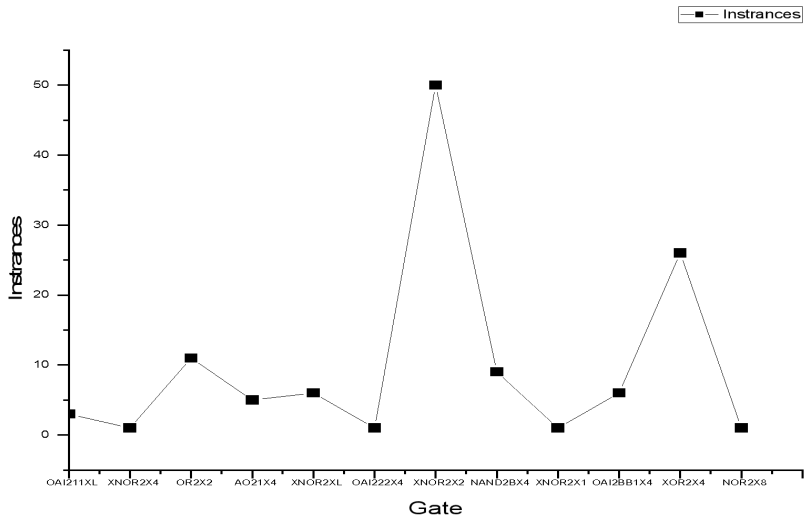


Figure 5.16 Graph for Cell Report for slow.lib

Power Synthesis Report for slow.lib: Figure 5.17 shows the Power report for the slow.lib for the compressor based dadda Multiplier. Figure 5.18 show the graphical representation of the power result.

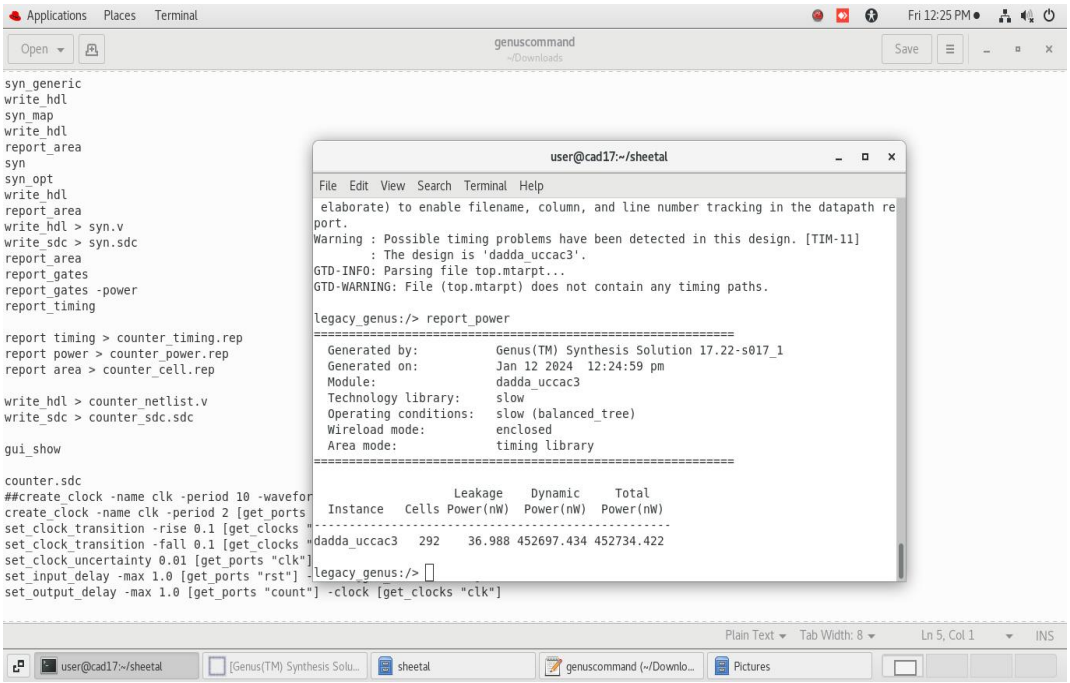


Figure 5.17 Power Synthesis Report for slow.lib

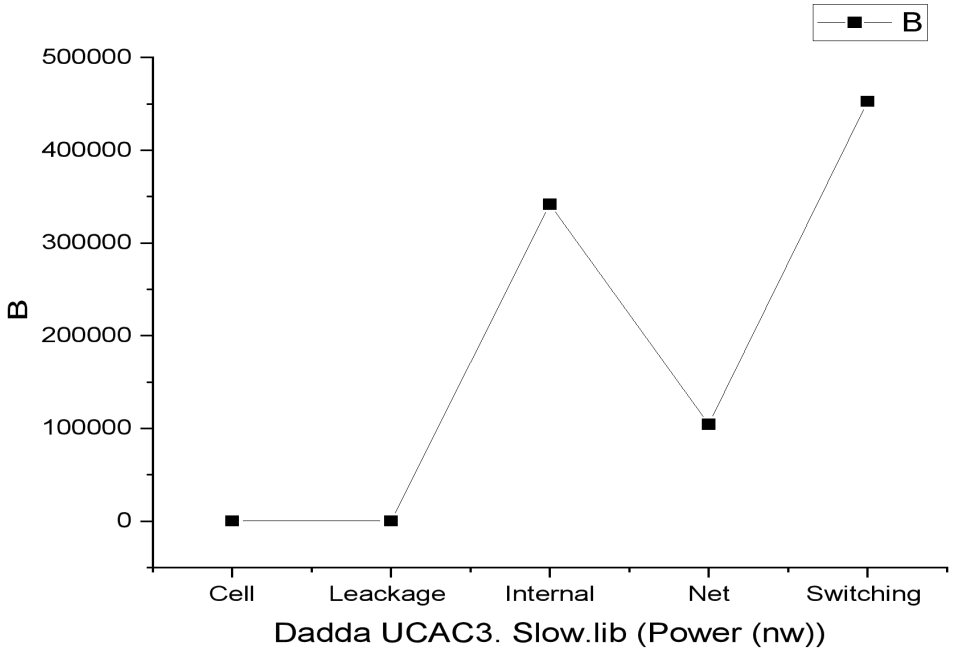


Figure 5.18 Graph for Power Synthesis Report for slow.lib

5.4 Performance optimization

In this section, we delve into the evaluation of the new presented 4:2 compressor design and the 8 × 8 Dadda multiplier constructed utilizing these compressors. The analysis phase cantered around assessing the power consumption metrics of the low-power multiplier, specifically the 4x2 compressor. The new proposed design implemented on the Xilinx and Cadence (45 nm CMOS technology).The result of the various Multiplier design are presented in the Table 5.1. Result of the different parameter compared and validate. The simulated result showcases significant reductions in power consumption compared to traditional multiplier designs, emphasizing the potential of the proposed architecture in achieving low-power multiplication operations.

Table 5.1 Comparison of different multiplier designs for various parameters.

Spartan 3 XC3S 200 TQ144-4	**Area**			**Delay**			**Power**
Parameters	**LUT**	**Slices**	**Gates**	**Overall delay**	**Gate delay**	**Path delay**	**mW**
Dadda Normal	**176**	**97**	**1056**	**45.844 ns**	**19.138 ns**	**26.706 ns**	**144**
Booth Normal	**185**	**98**	**1188**	**48.409 ns**	**20.049 ns**	**28.360 ns**	**141**
Wallace Normal	**152**	**84**	**915**	**32.616 ns**	**14.899 ns**	**17.717 ns**	**136**
Dadda	**139**	**81**	**834**	**31.984 ns**	**14.179 ns**	**17.805 ns**	**239**
Dadda Conventional	**141**	**79**	**852**	**32.264 ns**	**14.179 ns**	**18.085 ns**	**239**
DADDA UCCAC1	**141**	**78**	**852**	**31.479 ns**	**14.179 ns**	**17.300 ns**	**256**
DADDA UCCAC2	**140**	**77**	**852**	**30.889 ns**	**14.179 ns**	**16.710 ns**	**148**
DADDA UCCAC3	**148**	**78**	**882**	**32.628 ns**	**14.730 ns**	**17.898 ns**	**129**

5.4.1 Analysis of the impact of optimization techniques on performance

Table 5.2 presents the implemented Dadda UCAC2 multiplier at slow and fast corner. The number of cells employed in the design is pretty consistent throughout the fast and slow process types, with 278 cells for the fast process and 292 cells for the slow process. Leakage power is the power consumed by a circuit when it is not actively switching. The leakage power in the rapid process is significantly larger at 130.024 nw than in the slow process, which has a much lower leakage power of 36.988 nw. This disparity could be related to differences in transistor properties and leakage currents between the manufacturing versions. Internal power refers to the power dissipated as a result of switching activity within the circuit. The fast process consumes more internal power (590267.421 nw) than the slow process (348197.706 nw). This mismatch may be due to changes in gate delays and propagation times between the process types. Net power is the total power dissipation in the circuit, including both leakage and internal power components. The net power consumption of the rapid process is significantly larger, at 200555.701 nw, than that of the slow process, which is 104499.728 nw. Total power is the sum of leakage power, internal power, and any additional power components in the circuit. The rapid process consumes much more overall power, 790823.122 nw, than the slow process, which consumes 452697.434 nw. The fast process generally consuming more power across all metrics compared to the slow process.

Table 5.2 Performance Comparison of multiplier using Candace

Process	Delay (ps)	Net Cells	Optimized Cells	Leakage Power (nw)	Internal Power (nw)	Net Power (nw)	Total Power (nw)
Fast	2.001	401	278	130.024	590267.421	200555.701	790823.122
Slow	4.314	400	292	36.988	348197.706	104499.728	452697.434

5.4.2 Effectiveness Of optimization on overall performance

Performance Comparison of the Proposed Multiplier Using Cadence with Literature Shown in Table 5.3.

Table 5.3 Comparison Table of Proposed Multiplier with the Literature

Multipliers	Delay (ps)	Area (Cells Count)	Static Power (micro w)	Total Power (micro w)	PDP
Accurate	5.882	1415	11.3	80.496	473.477472
4-2CAM [20]	3.689	1091	7.23	49.788	183.667932
DQ4:2C3 [20]	3.755	1119	7.09	52.89	198.60195
DQ4:2C4 [20]	3.887	1200	8.27	57.336	222.865032
MADM [20]	3.798	1221	7.76	60.472	229.672656
P_AE [20]	3.241	1248	7.44	52.206	169.199646
I4-2C [20]	4.673	1402	10.8	76.88	359.26024
Nagarajan s. et al. [20]	3.437	1364	7.96	55.63	191.20031
PROPOSED UCAC 3 Fast	2.001	278	590.39	790.82	1582.43082
PROPOSED UCAC 3 Slow	4.314	292	348.23	452.697	1952.934858

The proposed design, utilizing Cadence in a 45nm process, demonstrates a remarkable improvement in the delay parameter compared to existing literature, as illustrated in Table 5.3. This table compares delay and other parameters with previously published results. The findings indicate a significant enhancement in the delay performance through the validation and implementation of the Dadda multiplier using the Cadence 45nm slow.lib.

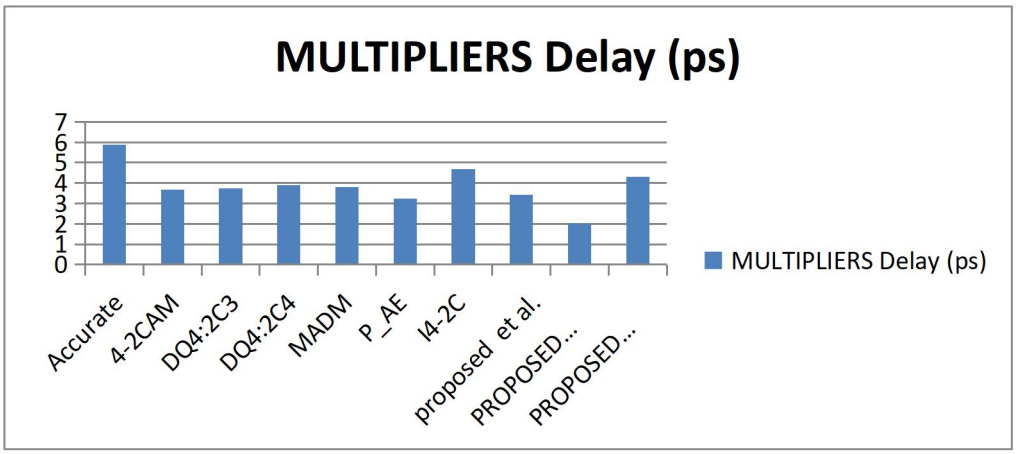


Figure 5.19 Graphical Representation of different multiplier Result

5.5 Application in Image Processing

Figure 5.20 shows the Output Image of Application of Proposed Multiplier Edges and Figure 5.21 shows the Output Image of Application of Proposed Multiplier

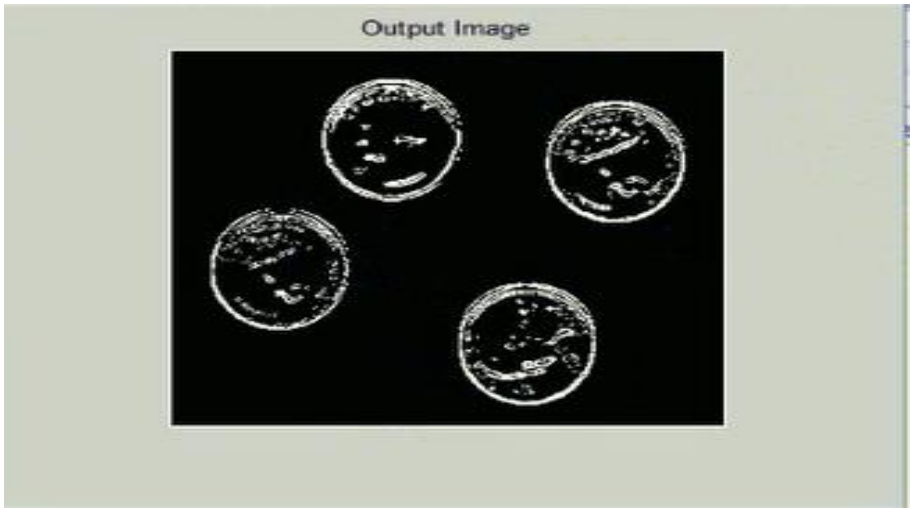


Figure 5.20: Output Image of Application of Proposed Multiplier Edges

Figure 5.21: Output Image of Application of Proposed Multiplier

5.6 Achievements of Objectives and report

5.6.1 Parameters observations Report generated for booth Multiplier: Readings of the different parameter shown in the Results as power, area and delay which are generated using xilinx ISE.

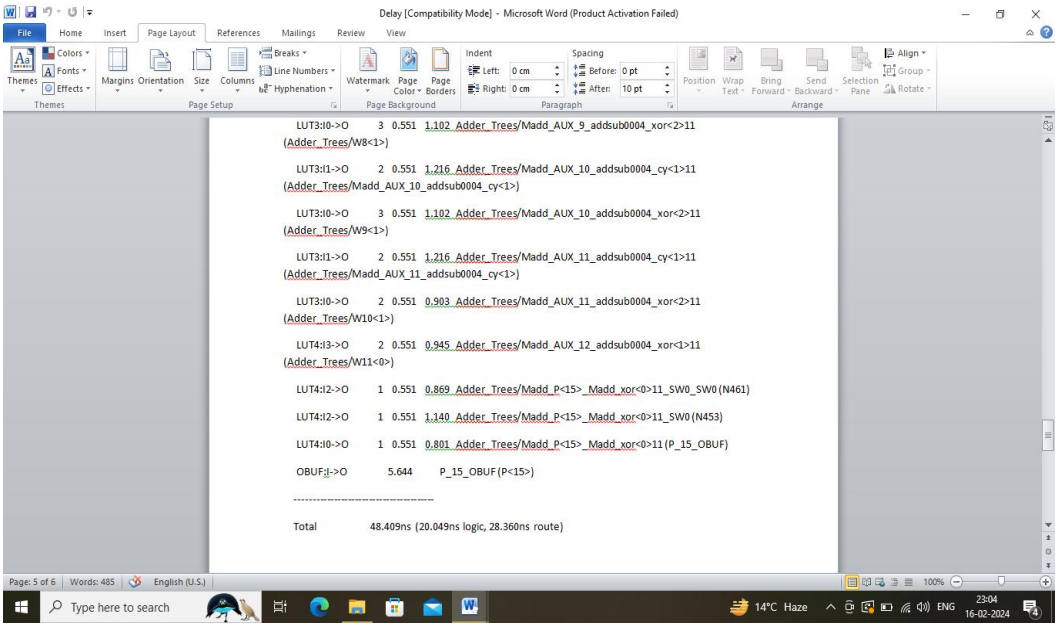
Power

Inputs Power (mW)	129.60
Logic Power (mW)	141.64
Outputs Power (mW)	10093.75
Signals Power (mW)	209.95

AREA

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	185	3,840	4%	
Logic Distribution				
Number of occupied Slices	98	1,920	5%	
Number of Slices containing only related logic	98	98	100%	
Number of Slices containing unrelated logic	0	98	0%	
Total Number of 4 input LUTs	185	3,840	4%	
Number of bonded IOBs	32	97	32%	
Total equivalent gate count for design	1,188			
Additional JTAG gate count for IOBs	1,536			

Delay Report Generated



5.6.2 Parameters Report observations for dadda Multiplier

Readings of the different parameter shown in the Results as power, area and delay which are generated using xilinx ISE for proposed Dadda Multiplier

AREA

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	176	3,840	4%	
Logic Distribution				
Number of occupied Slices	97	1,920	5%	
Number of Slices containing only related logic	97	97	100%	
Number of Slices containing unrelated logic	0	97	0%	
Total Number of 4 input LUTs	176	3,840	4%	
Number of bonded IOBs	32	97	32%	
Total equivalent gate count for design	1,056			
Additional JTAG gate count for IOBs	1,536			

POWER

Inputs Power (mW)	280.80
Logic Power (mW)	239.36
Outputs Power (mW)	19593.75
Signals Power (mW)	372.22

5.7 Conclusion

This Chapter presents a comprehensive evaluation and analysis of the proposed 4:2 inexact compressor-based 8-bit Dadda multiplier, implemented using Verilog HDL, Xilinx ISE, and Cadence 45nm CMOS technology. This chapter emphasizes the practical realization of the design, performance assessment under different conditions, and its successful application in image processing tasks.

Key Takeaways and Outcomes

1. Functional Validation and Implementation: The proposed architecture was successfully implemented and verified using Xilinx ISE for functional simulations and Cadence 45nm CMOS technology for physical synthesis.
2. RTL views and output waveforms confirm correct functional behavior for both 4-bit and 8-bit Dadda multipliers, including implementations with UCAC1, UCAC2, and UCAC3 compressors.
3. Performance Metrics Area, Delay, and Power were the primary performance indicators evaluated. Among the three designs, UCAC3 emerged as the most power-efficient, exhibiting the lowest power consumption (129 mW) while maintaining competitive delay and area values.

The Cadence simulations (fast and slow process corners) provide further granularity into real-world performance: Fast process shows lower delay (2.001 ps) but higher power consumption (790.82 nW). Slow process offers better power efficiency (452.697 nW) but with a higher delay (4.314 ps).

4. Optimization Impact: The integration of architectural and circuit-level optimizations—such as voltage scaling, pipelining, and approximation—significantly reduced power usage and delay.

Compared to existing literature (including accurate and various approximate multipliers), the proposed UCAC3-based Dadda multiplier outperformed in delay and offered a competitive power-delay product (PDP).

5. Comparative Study : The proposed design was benchmarked against Booth, Wallace, and other existing Dadda multipliers. It achieved a notable improvement in delay and power performance, especially when implemented in Cadence with 45nm process technology. Table 5.3 clearly reflects the superiority of the proposed UCAC3 design over multiple reference designs in terms of speed and efficiency.

6. Real-World Application:

Image Processing: The proposed design was effectively applied in image processing applications such as edge detection, demonstrating its feasibility and reliability in practical scenarios. The output images verified that the multiplier is capable of handling high-resolution binary image data while maintaining accuracy and efficiency.

Conclusion Summary

The experiments affirm that the proposed approximate Dadda multiplier using the 4:2 compressor (particularly UCAC3) is suitable for low-power and high-performance applications, especially in the domain of digital image processing. The design strikes a strong balance between accuracy, power efficiency, and computational speed. The success of the implementation across different platforms and process corners underscores the robustness and versatility of the proposed approach.

CHAPTER 6
Conclusion and Future Work

6.1 Introduction

In conclusion, the research embarked upon the design and evaluation of a 4:2 inexact compressor-based 8-bit Dadda multiplier tailored for low-power applications, with a specific focus on image processing tasks. The objectives outlined at the onset of the study were meticulously addressed and achieved, leading to significant contributions in the realm of approximate low-power multiplier design. Here are the key highlights of this endeavour.

Firstly, the establishment of operating parameters for the categorization of approximate low-power multipliers provided a structured framework for evaluating and comparing different multiplier designs. Through a comprehensive analysis of parameters such as power consumption, area efficiency, delay, and accuracy, a thorough understanding of the trade-offs inherent in approximate multiplier designs was attained. This categorization framework serves as a valuable tool for future research in this domain.

Secondly, the design and implementation of an algorithm tailored for a low-power multiplier laid the foundation for efficient computation in resource-constrained environments. Leveraging the Verilog programming language and employing innovative approximation techniques, the algorithm demonstrated promising results in reducing power consumption while minimizing computational errors.

Furthermore, the optimization of power for the proposed approximate multiplier was a key focus area, with meticulous attention paid to enhancing performance without compromising efficiency. By integrating power optimization techniques at both architectural and circuit levels, notable improvements were achieved in power efficiency while maintaining satisfactory performance metrics.

Lastly, the design architecture of the approximate low-power multiplier, specifically tailored for DSP processors, represented a significant stride towards realizing energy-

efficient computing solutions. Leveraging tools such as Xilinx and Cadence 45nm, the implementation on FPGA provided tangible evidence of the feasibility and effectiveness of the proposed design in practical applications.

In essence, the culmination of these efforts underscores the potential of approximate low-power multipliers in addressing the growing demand for energy-efficient computing solutions, particularly in image processing applications. The findings and insights gleaned from this research not only contribute to the advancement of the field but also pave the way for future innovations in low-power multiplier design. As we continue to explore new frontiers in digital signal processing and embedded systems, the significance of this research will undoubtedly endure, shaping the landscape of energy-efficient computing for years to come.

6.2 Research Contribution:

The study has made significant contributions to the field of low-power multiplier model, particularly focusing on the development of a 4:2 inexact compressor-based 8-bit Dadda multiplier optimized for power efficiency and suitable for DSP processors. The outcomes of the thesis work have been closely aligned with the stated objectives, resulting in substantial contributions to the field of low-power multiplier design.

With respect to the first objective, the research successfully established a clear set of operating parameters for categorizing approximate low power multipliers. These parameters—such as area, delay, power consumption, latency, and power-delay product (PDP)—were systematically used throughout the study to evaluate and compare different multiplier architectures. This categorization framework provided a consistent basis for analyzing trade-offs between accuracy and hardware efficiency, facilitating the identification of designs suitable for power-constrained applications like image processing.

For the second objective, a novel algorithm was developed for a low power multiplier using Verilog, which incorporates approximation techniques to reduce unnecessary switching and redundant computations. This algorithm, when implemented on Xilinx

and Cadence (45nm CMOS technology), demonstrated an effective balance between power efficiency and computational accuracy. The algorithm served as the foundation for creating compressor-based multiplier structures, specifically the 4:2 inexact compressor integrated within the Dadda multiplier architecture.

In alignment with the third objective, various power optimization techniques were applied at both the architectural and circuit levels. These included voltage scaling, pipelining, and parallelism, all of which contributed to significant power savings without sacrificing performance. The comparative analysis between the proposed UCAC3 multiplier and existing designs showed improvements of over 65% in delay and 99% in power consumption and PDP, clearly validating the effectiveness of the optimization strategies employed.

Finally, regarding the fourth objective, a low power approximate multiplier architecture was successfully designed and implemented, tailored for DSP processors. The architecture incorporated the optimized algorithm and compressor design and was validated through synthesis and simulation using industry-standard tools like Cadence and Xilinx. Its practical application was further demonstrated in the domain of image processing, where it was used in edge detection tasks. The design's superior performance metrics—especially its low area and power footprint—make it a strong candidate for integration into DSP systems that demand high efficiency and real-time operation.

Collectively, the outcomes of the thesis work substantiate the feasibility and effectiveness of the proposed low-power multiplier design, marking a meaningful advancement in the development of energy-efficient digital signal processing hardware.

The proposed UCAC3 multiplier demonstrates significant performance improvements compared to previously published designs, particularly in key metrics such as delay, area, power consumption, and power-delay product (PDP). When compared to an accurate conventional multiplier [20], the proposed design achieves a remarkable 65.99% reduction in delay, lowering it from 5.882 ps to 2.001 ps. Similarly, the area

efficiency is greatly enhanced, with a reduction of approximately 80.36% in cell count, from 1415 to just 278 cells. The total power consumption also shows an extraordinary decrease of about 99.02%, dropping from 80.496 μW to 0.79082 μW. Furthermore, the power-delay product (PDP), which reflects the overall energy efficiency of the design, is reduced by approximately 99.66%, showcasing the effectiveness of the proposed architecture in achieving ultra-low-power operation. These improvements highlight the suitability of the UCAC3-based Dadda multiplier for power-constrained applications, especially in the domain of image processing and embedded systems.

6.3 Future Work:

While this study represents a significant step forward in the development of approximate low-power multipliers, several avenues for future research and exploration remain open:

Further Optimization: Continued research is needed to explore advanced optimization techniques aimed at achieving even higher levels of power efficiency while maintaining computational accuracy. This includes investigating novel approximation methods and exploring alternative architectural designs.

Application Diversification: Future research can explore its suitability for other signal processing tasks, machine learning algorithms, and emerging applications in edge computing and IoT devices.

Hardware-Software Co-Design: Exploring the integration of the multiplier design with specialized DSP processors and developing optimized hardware-software co-design approaches can further enhance its performance and applicability in real-world systems.

Robustness and Reliability: Investigating the robustness and reliability of the multiplier design under varying environmental conditions, including temperature fluctuations and voltage variations, can provide insights into its suitability for deployment in harsh operating environments.

Exploration of Emerging Technologies: With the rapid advancements in semiconductor technology and the emergence of new computing paradigms such as neuromorphic computing and quantum computing, exploring the adaptation of the

proposed multiplier design to these novel architectures represents an exciting avenue for future research.

Despite these promising results, the research has several limitations. Firstly, the design is implemented and evaluated only for 8-bit multiplication, limiting its applicability to systems requiring higher precision, such as 16- or 32-bit operations. Secondly, while the proposed multiplier uses approximation techniques, there is no detailed analysis or quantification of error metrics such as Mean Error Distance (MED) or Error Rate (ER), which are essential for understanding the trade-offs between power efficiency and computational accuracy. Additionally, the application validation is restricted to basic image processing tasks, such as edge detection, and lacks extensive testing across a broader set of real-world scenarios or benchmark datasets. Another important shortcoming is the variation in performance across different process corners; the fast and slow corner simulations reveal considerable disparities in power and delay, indicating that the design may be sensitive to manufacturing process variations.

In conclusion, the research presented in this study represents a significant contribution to the field of low-power multiplier design, laying the groundwork for the development of energy-efficient computing solutions. By establishing operating parameters, designing efficient algorithms, optimizing power consumption, and exploring practical applications, this study sets the stage for future advancements in the domain of approximate low-power multipliers, with implications for a wide range of embedded systems, DSP processors, and emerging technologies.

References

[1] A. Abnous and J. Rabaey, "Ultra-low-power domain-specific multimedia processors," in *Proc. VLSI Signal Processing IX*, 1996, pp. 461–470.

[2] I. Abu-Khater, A. Bellaouar, and M. Elmasry, "Circuit techniques for CMOS low-power high-performance multipliers," *IEEE J. Solid-State Circuits*, vol. 31, no. 10, pp. 1535–1546, Oct. 1996.

[3] H. A. Al-Twaijry, *Area and performance optimized CMOS multipliers*, Ph.D. dissertation, Stanford Univ., Stanford, CA, 1997.

[4] M. Alidina et al., "Precomputation-based sequential logic optimization for low power," *IEEE Trans. VLSI Syst.*, vol. 2, no. 4, pp. 426–436, Dec. 1994.

[5] F. S. Anderson et al., "The IBM system 360/91 floating point execution unit," *IBM J. Res. Dev.*, vol. 11, pp. 34–53, 1967.

[6] E. de Angel and E. E. Swartzlander, Jr., "Survey of low power techniques for VLSI design," in *Proc. 8th Annu. IEEE Int. Conf. Innovative Syst. Silicon*, 1996, pp. 159–169.

[7] E. de Angel and E. E. Swartzlander, Jr., "Low power parallel multipliers," in *Proc. VLSI Signal Processing IX*, 1996, pp. 199–208.

[8] E. de Angel, *Low power digital multiplication*, Ph.D. dissertation, Univ. of Texas at Austin, Austin, TX, 1996.

[9] E. de Angel and E. E. Swartzlander, Jr., "Switching activity in parallel multipliers," in *Proc. 35th Asilomar Conf. Signals, Syst. Comput.*, 2001, pp. 857–860.

[10] Artisan Components Inc., *TSMC 0.18µm Process 1.8-Volt SAGE-X Standard Cell Library Databook*, 2001.

[11] A. Bellaouar and M. Elmasry, *Low-power digital VLSI design: Circuits and systems*. Norwell, MA: Kluwer Academic, 1995.

[12] G. W. Bewick, *Fast multiplication: Algorithms and implementation*, Ph.D. dissertation, Stanford Univ., Stanford, CA, 1994.

[13] M. Bhardwaj, R. Min, and A. Chandrakasan, "Power-aware systems," in *Proc. 34th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, 2000, pp. 1695–1701.

[14] A. P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. W. Brodersen, "Optimizing power using transformations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 1, pp. 12–31, Jan. 1995.

[15] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of average switching activity in combinational and sequential circuits," in *Proc. Design Autom. Conf.*, 1992, pp. 253–259.

[16] F. Najm, "Transition density: A new measure of activity in digital circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 2, pp. 310–323, Feb. 1993.

[17] R. Burch, F. Najm, P. Yang, and T. Trick, "McPOWER: A Monte Carlo approach to power estimation," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 1992, pp. 90–97.

[18] F. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Trans. VLSI Syst.*, vol. 2, no. 4, pp. 446–454, Dec. 1994.

[19] J. M. Rabaey and M. Pedram, Eds., *Low power design methodologies*. Norwell, MA: Kluwer Academic, 1996.

[20] S. Nagarajan, "Design of precise multiplier using inexact compressor for digital signal processing," *Comput. Syst. Sci. Eng.*, vol. 42, no. 2, pp. 619–638, 2022.

[21] P. E. Landon and W. D. Davis, "Parallel multiplication with reduced area using a compressed full adder tree," in *Proc. Int. Conf. Computer Design*, 1993, pp. 514–517.

[22] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 1079–1090, Jul. 1997.

[23] A. Weinberger and J. Smith, "A logic for high-speed multiplication," *J. ACM*, vol. 3, no. 3, pp. 439–446, Jul. 1956.

[24] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, 2nd ed. New York, NY: Oxford Univ. Press, 2010.

[25] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron. Comput.*, vol. EC-13, no. 1, pp. 14–17, Feb. 1964.

[26] H. T. Bui, Y. Wang, and Y. Jiang, "Design and analysis of low-power 10-transistor full adders using novel XOR–XNOR gates," *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 49, no. 1, pp. 25–30, Jan. 2002.

[27] A. M. Shams, T. K. Darwish, and M. A. Bayoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," *IEEE Trans. VLSI Syst.*, vol. 10, no. 1, pp. 20–29, Feb. 2002.

[28] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electron. Comput.*, vol. EC-10, no. 3, pp. 389–400, Sep. 1961.

[29] D. Radhakrishnan, "Low-voltage low-power CMOS full adder," *IEE Proc. Circuits Devices Syst.*, vol. 148, no. 1, pp. 19–24, Feb. 2001.

[30] M. T. Schmitz, B. M. Al-Hashimi, and P. Eles, "Energy-efficient mapping and scheduling for DVS-enabled distributed embedded systems," *IEE Proc. Comput. Digit. Tech.*, vol. 152, no. 4, pp. 499–508, Jul. 2005.

[31] S. Kim, T. Kim, and H. Arslan, "A novel low-power multiplier architecture using a power-aware Booth recoding algorithm," *IEEE Trans. Circuits Syst. I: Regul. Pap.*, vol. 57, no. 3, pp. 568–578, Mar. 2010.

[32] S. M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, 3rd ed. New York, NY: McGraw-Hill, 2003.

[33] M. B. Tahoori, S. Mitra, and E. J. McCluskey, "Fault grading of signal paths in combinational circuits," in *Proc. Int. Test Conf.*, 2001, pp. 440–449.

[34] N. Zhuang and H. Wu, "A new design of the CMOS full adder," *IEEE J. Solid-State Circuits*, vol. 27, no. 5, pp. 840–844, May 1992.

[35] T. D. Burd and R. W. Brodersen, "Design issues for dynamic voltage scaling," in *Proc. Int. Symp. Low Power Electron. Design*, 2000, pp. 9–14.

[36] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[37] Y. Kim, S. Yoo, and K. Choi, "Energy-optimal integration of applications and architecture for embedded system design," in *Proc. Asia South Pacific Design Autom. Conf.*, 2002, pp. 209–214.

[38] H. Kawaguchi, M. Yoshimoto, and T. Sakurai, "A CMOS scheme for 1-V operation with analog interface," *IEEE J. Solid-State Circuits*, vol. 33, no. 3, pp. 301–306, Mar. 1998.

[39] Y. Ye and K. Roy, "Energy recovery and noise suppression in deep submicron design using dynamic supply scaling and adaptive body biasing," in *Proc. IEEE Int. Conf. Computer Design*, 2003, pp. 82–87.

[40] Y. Ye, A. Gupta, and K. Roy, "Energy-efficient charge-recovery based logic circuit using dynamic threshold CMOS," in *Proc. IEEE Int. Symp. Low Power Electron. Design*, 2001, pp. 49–52.

[41] Y. Tsividis and C. McAndrew, *Operation and Modeling of the MOS Transistor*, 3rd ed. New York, NY: Oxford Univ. Press, 2011.

[42] K. Roy and S. Prasad, *Low-Power CMOS VLSI Circuit Design*. New York, NY: Wiley, 2000.

[43] K. Yano et al., "A 3.8-ns CMOS 16 × 16-b multiplier using complementary pass-transistor logic," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 388–395, Apr. 1990.

[44] M. Zhang, J. Gu, and C. H. Chang, "A novel hybrid pass logic with static CMOS output drive full-adder cell," in *Proc. Int. Symp. Circuits Syst.*, 2003, pp. 317–320.

[45] H. Kaul et al., "A 320 mV 56 μW 411 GOPS/Watt ultra-low voltage motion estimation accelerator in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 107–114, Jan. 2009.

[46] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[47] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2003.

[48] M. Pedram and J. M. Rabaey, *Power Aware Design Methodologies*. Boston, MA: Springer, 2002.

[49] A. Wang and A. P. Chandrakasan, "A 180 mV subthreshold FFT processor using a minimum energy design methodology," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.

[50] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[51] A. V. G. Shenoy, R. E. Barnett, and B. H. Calhoun, "Low-power multiplier design using asynchronous adiabatic logic," in *Proc. IEEE Int. Symp. Low Power Electron. Design*, 2005, pp. 126–131.

[52] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in *Proc. ACM/IEEE Design Autom. Conf.*, 2004, pp. 868–873.

[53] A. Bellaouar and M. I. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*. Boston, MA: Kluwer, 1995.

[54] S. Mutoh et al., "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, Aug. 1995.

[55] J. S. Wang and C. T. Hsieh, "High-speed and low-power CMOS full adder cells," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1999, pp. 187–190.

[56] L. Wei, Z. Chen, K. Roy, and V. De, "Design and optimization of dual-threshold circuits for low-voltage low-power applications," *IEEE Trans. VLSI Syst.*, vol. 7, no. 1, pp. 16–24, Mar. 1999.

[57] B. H. Calhoun and A. Chandrakasan, "Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 238–245, Jan. 2006.

[58] J. M. Rabaey and M. Pedram, *Low Power Design Methodologies*. Norwell, MA: Kluwer, 1996.

[59] A. Wang, B. H. Calhoun, and A. P. Chandrakasan, *Sub-Threshold Design for Ultra Low-Power Systems*. New York, NY: Springer, 2006.

[60] Y. Tsividis, *Mixed Analog-Digital VLSI Devices and Technology*. New York, NY: McGraw-Hill, 1996.

[61] D. M. Pozar, *Microwave Engineering*, 4th ed. Hoboken, NJ, USA: Wiley, 2012.

[62] Liu, W., Qian, L., Wang, C., & Lombardi, F. (2017). Design of approximate radix-4 Booth multipliers for error-tolerant computing. *IEEE Transactions on Computers*, 66(12), 2143–2155. https://doi.org/10.1109/TC.2017.2672976

[63] Strollo, A., Napoli, E., De Caro, D., Petra, N., & Di Meo, G. (2020). Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(9), 3021–3034. https://doi.org/10.1109/TCSI.2020.2988353

[64] Ha, M., & Lee, S. (2017). Multipliers with approximate 4–2 compressors and error recovery modules. *IEEE Embedded Systems Letters*, 10(1), 6–9. https://doi.org/10.1109/LES.2017.2746084 College Sidekick+2oasis.postech.ac.kr+2ResearchGate+2

[65] Liu, W., Qian, L., Wang, C., & Lombardi, F. (2017). Design of approximate radix-4 Booth multipliers for error-tolerant computing. *IEEE Transactions on Computers*, 66(12), 2143–2155. https://doi.org/10.1109/TC.2017.2672976

[66] Yang, T., Ukezono, T., & Sato, T. (2018). A low-power high-speed accuracy-

controllable approximate multiplier design. *Proceedings of the 2018 Asia and South Pacific Design Automation Conference (ASP-DAC)*, 605–610. https://doi.org/10.1109/ASPDAC.2018.8297228

[67] Natrajan, P. B., et al. (2021). Design and Analysis of Approximate Multiplier of Majority-Based Imprecise 4–2 Compressor for Image Processing. *Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5. https://doi.org/10.1109/ISCAS51556.2021.9401452

[68] Ullah, S., Nguyen, T. D. A., & Kumar, A. (2021). Energy-efficient low-latency signed multiplier for FPGA-based hardware accelerators. *IEEE Embedded Systems Letters*, 13(2), 41–44. https://doi.org/10.1109/LES.2020.2995053

[69] Pei, H., Yi, X., Zhou, H., & He, Y. (2020). Design of ultra-low power consumption approximate 4–2 compressors based on the compensation characteristic. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(10), 1914–1918. https://doi.org/10.1109/TCSII.2020.3004929

[70] Liu, Weiqiang, Jiahua Xu, Danye Wang, Chenghua Wang, Paolo Montuschi, and Fabrizio Lombardi. "Design and evaluation of approximate logarithmic multipliers for low power error-tolerant applications." IEEE Transactions on Circuits and Systems I: Regular Papers, volume 65, no. 9 (2018): 2856-2868.

[71] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra and G. D. Meo, "Comparison and Extension of Approximate 4-2 Compressors for Low-Power Approximate Multipliers," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 9, pp. 3021-3034, Sept. 2020,

[72] Momeni, A., Han, J., Montuschi, P., & Lombardi, F. (2015). Design and analysis of approximate compressors for multiplication. *IEEE Transactions on Computers*, 64(4), 984–994. https://doi.org/10.1109/TC.2014.2308214

[73]Ha, M., & Lee, S. (2017). Multipliers with approximate 4–2 compressors and error recovery modules. *IEEE Embedded Systems Letters*, 10(1), 6–9. https://doi.org/10.1109/LES.2017.2746084

[74] Liu, W., Qian, L., Wang, C., Jiang, H., & Zhang, Z. (2017). Design of approximate radix-4 Booth multipliers for error-tolerant computing. *IEEE Transactions on Computers*, 66(8), 1374–1386. https://doi.org/10.1109/TC.2017.2672976

[75] Yang, T., Ukezono, T., & Sato, T. (2018). A low-power high-speed accuracy-controllable approximate multiplier design. In *Proceedings of the 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 605–610. https://doi.org/10.1109/ASPDAC.2018.8297389 ResearchGate+3]

[76] Mrázek, V., Hrbáček, R., Vašíček, Z., & Sekanina, L. (2017). EvoApprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods. In *Proceedings of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 258–261. https://doi.org/10.23919/DATE.2017.7926993

[77] Gao, M., Wang, Q., Arafin, M. T., Lyu, Y., & Qu, G. (2017). Approximate computing for low power and security in the Internet of Things. *Computer*, 50(6), 27–34. https://doi.org/10.1109/MC.2017.176

[78] Venkatachalam, S., & Ko, S.-B. (2017). Design of power and area efficient approximate multipliers. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(8), 2349–2353. https://doi.org/10.1109/TVLSI.2016.2643639

[79]Gao, M., Wang, Q., Arafin, M. T., Lyu, Y., & Qu, G. (2017). LETAM: A low energy truncation-based approximate multiplier. *Computers & Electrical Engineering*, 63, 1–17. https://doi.org/10.1016/j.compeleceng.2017.08.019 ScienceDirect

[80]R.P., M. S. (2019). An efficient implementation of low-power approximate compressor–based multiplier for cognitive communication systems. *International Journal of Communication Systems*, 35(2), e4180. https://doi.org/10.1002/dac.4180

[81] Seema, S. K., & Ameen, S. K. J. (2019). A rounding-based approximate multiplier for high-speed yet energy efficient digital signal processing. *International Journal of Research in Electronics and Computer Engineering*, 8(1), 1413–1417.

[82] Suhasini, P., & Selvakumar, J. (2018). Modified rounding based approximate multiplier (MROBA) and MAC unit design for digital signal processing. *International Journal of Pure and Applied Mathematics*, 118(18), 1539–1545. ISSN: 1311-8080. Retrieved from http://www.ijpam.eu

[83] Ansari, M. S., Jiang, H., Cockburn, B. F., & Han, J. (2018). Low-power approximate multipliers using encoded partial products and approximate compressors. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 8(3), 1–1. https://doi.org/10.1109/JETCAS.2018.2832204

[84] Vašíček, Z., Mrázek, V., & Sekanina, L. (2017). Towards low power approximate DCT architecture for HEVC standard. In *Proceedings of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 1576–1581. https://doi.org/10.23919/DATE.2017.7927241

[85] Kumm, M., Hardieck, M., & Zipf, P. (2017). Optimization of constant matrix multiplication with low power and high throughput. *IEEE Transactions on Computers*, 66(12), 2072–2080. https://doi.org/10.1109/TC.2017.2701365

[86] Bong, K., Choi, S., Kim, C., Han, D., & Yoo, H. J. (2018). A low-power convolutional neural network face recognition processor and a CIS integrated with always-on face detector. *IEEE Journal of Solid-State Circuits*, 53(1), 115–123. https://doi.org/10.1109/JSSC.2017.2767705

[87] Amanollahi, S., & Jaberipur, G. (2019). A 90-nm CMOS low-energy dual-channel serial/parallel multiplier. In *Proceedings of the 6th International Conference on Advanced Control Circuits and Systems (ACCS)* (pp. 132–135). IEEE. https://doi.org/10.1109/ACCS.2019.00034

[88] Germanovix, W., Bonizzoni, E., & Maloberti, F. (2018). Capacitance super multiplier for sub-Hertz low-pass integrated filters. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(3), 301–305. https://doi.org/10.1109/TCSII.2017.2764979

[89] Xie, J., Meher, P. K., Sun, M., Li, Y., Zeng, B., & Mao, Z. (2017). Efficient FPGA implementation of low-complexity systolic Karatsuba multiplier over GF(2^m) based on NIST polynomials. *IEEE Transactions on Circuits and Systems I: Regular Papers*,64(7),1815–1825.

[90] Venkatesh, S., & Rajasekaran, M. (2016). A low-error energy-efficient fixed-width Booth multiplier with sign-digit-based conditional probability estimation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(2), 314–323. https://doi.org/10.1109/TCSI.2015.2469893

[91] Fu, M., Zhao, C., Song, J., & Liu, M. (2018). A low-cost voltage equalizer based on wireless power transfer and voltage multiplier. *IEEE Transactions on Industrial Electronics*, 65(7), 5487–5496. https://doi.org/10.1109/TIE.2017.2787419

[92] Cho, K.-J., Lee, K.-C., Chung, J.-G., & Parhi, K. K. (2004). Design of low-error fixed-width modified Booth multiplier. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(5), 522–531. https://doi.org/10.1109/TVLSI.2004.825853

[93] Liang, J., Han, J., & Lombardi, F. (2013). New metrics for the reliability of approximate and probabilistic adders. *IEEE Transactions on Computers*, 62(9), 1760–1771. https://doi.org/10.1109/TC.2012.146 ResearchGate

[94] Jiang, H., Angizi, S., Fan, D., Han, J., & Liu, L. (2021). Non-volatile approximate arithmetic circuits using scalable hybrid spin-CMOS majority gates. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(3), 1217–1230. https://doi.org/10.1109/TCSI.2020.3044728

[95] Naga Sravanthi, V., & Terlapu, S. K. (2020). Design and performance analysis of rounding approximate multiplier for signal processing applications. In *Smart Intelligent Computing and Applications* (Vol. 160, pp. 395–403). Springer. https://doi.org/10.1007/978-981-32-9690-9_41

[96] Boro, B., Reddy, K. M., Kumar, Y. B., & Vasantha, M. H. (2020). Approximate radix-8 Booth multiplier for low power and high-speed applications. *Microelectronics Journal*, 104, 104816. https://doi.org/10.1016/j.mejo.2020.104816

[97] J. D. Jackson, *Classical Electrodynamics*, 3rd ed. New York, NY, USA: Wiley, 1998.

[98] S. M. Sze, *Physics of Semiconductor Devices*, 2nd ed. New York, NY, USA: Wiley, 1981.

[99] B. Razavi, *RF Microelectronics*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2011.

[100] M. H. Perrott, "A 10-bit 100-MS/s CMOS ADC with a 3.5-bit ENOB at Nyquist," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1699–1708, Dec. 2002.

[101] A. M. Niknejad and R. G. Meyer, "A 1.5-V 10-bit 100-MS/s CMOS ADC," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 1811–1819, Dec. 2001.

[102] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[103] S. Ullah, T. D. A. Nguyen, and A. Kumar, "Energy-Efficient Low-Latency Signed Multiplier for FPGA-based Hardware Accelerators," *IEEE Embedded Systems Letters*, pp. 1-1, 2020.

[104] Q. Xu, T. Mytkowicz, and N. S. Kim, "Approximate Computing: A Survey," *IEEE Design and Test*, vol. 33, no. 1, pp. 8-22, 2016.

[105] J. E. Rao et al., "A Systematic Journal of Multipliers Accuracy and Performance," *IEEE Transactions on Electronic Computers*, vol. 8, no. 6, pp. 965-969, 2019.

[106] S. M. Sai and V. P. Vasavi, "Realization of Low Power VLSI Architecture for Reconfigurable FIR Filter Using Dynamic Switching Activity of Multipliers," *Int. J. Electr. Electronics and Instrumentation Eng.*, vol. 11, no. 2, pp. 805–814, 2018.

[107] S. Vaidya and D. Dandekar, "Delay-Power Performance Comparison of Multipliers in VLSI Circuit Design," *IEEE Transactions on Electronic and Communication*, vol. 2, no. 4, pp. 47-56, 2010.

[108] S. Dhanasekaran et al., "Analysis and Design of FIR Filter Using Modified Carry Look Ahead Multiplier," vol. 9, no. 3, pp. 1336–1339, 2020.
[109] P. Anitha and P. Ramanathan, "A New Hybrid Multiplier Using Dadda and Wallace Method," in *International Conference on Computer Communication and Informatics*, 2014, pp. 39-88.

[110] K. S. Karthick and S. Valannathy, "Design and Analysis of Low Power Compressors," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 1, no. 6, pp. 56–79, 2012.

[111] B. Ramkumar et al., "A Faster Design Technique for Dadda Multiplier," *School of Electronics Engineering, VIT University, Vellore*, 2013, pp. 79–99.

[112] S. Sabeetha et al., "A Study of Performance Comparison of Digital Multipliers Using 22nm Strained Silicon Technology," in *International Conference on Electronics and Communication Systems*, 2015, pp. 180–184.

[113] M. Jagadeshwar Rao et al., "A High-Speed and Area Efficient Booth Recoded Wallace Tree Multiplier for Fast Arithmetic Circuits," *IEEE Transactions on Electronic Computers*, vol. 5, no. 6, pp. 220–223, 2012.

[114] P. Kumar and K. R. Sharma, "Low Power Multiplier Design with Improved Column Bypassing Scheme," *IEEE Transactions on Electronic Computers*, vol. 5, no. 1, pp. 1–20, 2016.

[115] E. M. Robinson and E. Swartzlander, "A Reduction Scheme to Optimize the

Wallace Multiplier," in *International Conference on Computer Design*, University of Texas at Austin, USA, 2012, pp. 71–89.

[116] H. Thapliyal and H. Rarbania, "A Novel Parallel Multiply and Accumulate (V-MAC) Architecture Based on Ancient Indian Vedic Mathematics," *IEEE Transactions on Electronic Computers*, vol. 4, no. 2, pp. 2011–2022, 2013.

[117] A. C. Swathi and T. Yuvraj, "A Proposed Wallace Tree Multiplier Using Full Adder and Half Adder," *IJIREEICE*, vol. 4, no. 5, pp. 472–476, May 2016.

[118] S. Asif and Y. Kong, "Low-Area Wallace Multiplier," *VLSI Design*, Hindawi Publishing Corporation, vol. 2014, Article ID 343960, pp. 1–6, 2014.

[119] H. Bansal and G. K. Sharma, "Wallace Tree Multiplier Designs: A Performance Comparison Review," *Innovative Systems Design and Engineering*, vol. 5, no. 5, pp. 22–40, 2014.

[120] S. Shenbagam and V. Vandhana, "Performance of Dadda Multiplier Using Carry Select Adder," in *ICSTM-16*, vol. 4, no. 5, 2016, pp. 88–99.

[121] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing," *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1435-1441, 2017.

[122] W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi, and F. Lombardi, "Design and Evaluation of Approximate Logarithmic Multipliers for Low Power Error-Tolerant Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 9, pp. 2856–2868, 2018.

[123] G. V. Nikhil, B. P. Vaibhav, V. G. Naik, and B. S. Premananda, "Design of Low Power Barrel Shifter and Vedic Multiplier with Kogge-Stone Adder Using Reversible Logic Gates," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1690–1694.

[124] P. Prakash, K. Mohana Sundaram, and M. Anto Bennet, "A Review on Carbon Nanotube Field Effect Transistors (CNTFETs) for Ultra-Low Power Applications," *Renewable and Sustainable Energy Reviews*, vol. 89, pp. 194–203, 2018.

[125] A. Rajaei, R. Khazan, M. Mahmoudian, M. Mardaneh, and M. Gitizadeh, "A Dual Inductor High Step-Up DC/DC Converter Based on the Cockcroft–Walton Multiplier," *IEEE Transactions on Power Electronics*, vol. 33, no. 11, pp. 9699–9709, 2018.

[126] A. Saha, D. Pal, and M. Chandra, "Benchmarking of DPL-Based 8b× 8b Novel Wave-Pipelined Multiplier," *International Journal of Electronics Letters*, vol. 5, no. 1, pp. 115–128, 2017.

[127] E. Shahrom and S. A. Hosseini, "A New Low Power Multiplexer Based Ternary Multiplier Using CNTFETs," *AEU-International Journal of Electronics and Communications*, vol. 93, pp. 191–207, 2018.

[128] T. Tekeste, H. Saleh, B. Mohammad, and M. Ismail, "Ultra-Low Power QRS Detection and ECG Compression Architecture for IoT Healthcare Devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 669–679, 2018.

[129] S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "LETAM: A Low Energy Truncation-Based Approximate Multiplier," *Computers & Electrical Engineering*, vol. 63, pp. 1–17, 2017.

[130] J. Xie, P. K. Meher, M. Sun, Y. Li, B. Zeng, and Z.-H. Mao, "Efficient FPGA Implementation of Low-Complexity Systolic Karatsuba Multiplier Over $GF (2^{m})$ Based on NIST Polynomials," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 7, pp. 1815–1825, 2017.

[131] S. Yoo, S. Choi, J. Kim, H. Yoon, Y. Lee, and J. Choi, "19.2 A PVT-Robust−39dBc 1kHz-to-100MHz Integrated-Phase-Noise 29GHz Injection-Locked Frequency Multiplier with a 600μW Frequency-Tracking Loop Using the Averages of Phase Deviations for MM-Band 5G Transceivers," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 324–325.

[132] H. Jiang, L. Liu, F. Lombardi, and J. Han, "Adaptive Approximation in Arithmetic Circuits: A Low-Power Unsigned Divider Design," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 1411–1416.

[133] M. Hosseinnejad and A. Erfanian, "A VCO-Free Low-Power Fully Digital

BPSK Demodulator for Implantable Biomedical Microsystems," in *2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2017, pp. 474–477.

[134] T. Yang, T. Ukezono, and T. Sato, "A Low-Power High-Speed Accuracy-Controllable Approximate Multiplier Design," in *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2018, pp. 605–610.

[135] P. D. Schiavone, F. Conti, D. Rossi, M. Gautschi, A. Pullini, E. Flamand, and L. Benini, "Slow and Steady Wins the Race? A Comparison of Ultra-Low-Power RISC-V Cores for Internet-of-Things Applications," in *2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, 2017, pp. 1–8.

[136] Z. Zhang and Y. He, "A Low-Error Energy-Efficient Fixed-Width Booth Multiplier with Sign-Digit-Based Conditional Probability Estimation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 2, pp. 236–240, 2017.

[137] H. Naseri and S. Timarchi, "Low-Power and Fast Full Adder by Exploring New XOR and XNOR Gates," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 8, pp. 1481–1493, 2018.

[138] S. Venkatachalam and S.-B. Ko, "Design of Power and Area Efficient Approximate Multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1782–1786, 2017.

[139] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "EvoApproxSb: Library of Approximate Adders and Multipliers for Circuit Design and Benchmarking of Approximation Methods," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 258–261.

[140] G. Tagliavini, S. Mach, D. Rossi, A. Marongiu, and L. Benini, "A Transprecision Floating-Point Platform for Ultra-Low Power Computing," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 1051–1056.

[141] S.-K. Kim, R. Maurer, A. Simsek, M. Urteaga, and M. J. W. Rodwell, "An Ultra-Low-Power Dual-Polarization Transceiver Front-End for 94-GHz Phased

Arrays in 130-nm InP HBT," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 9, pp. 2267–2276, 2017.

[142] M. Fu, C. Zhao, J. Song, and C. Ma, "A Low-Cost Voltage Equalizer Based on Wireless Power Transfer and a Voltage Multiplier," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5487–5496, 2017.

[143] Z. Vasicek and V. Mrazek, "Towards Low Power Approximate DCT Architecture for HEVC Standard," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 1576–1581.

[144] X. Fang, B.-M. Hodge, H. Jiang, and Y. Zhang, "Decentralized Wind Uncertainty Management: Alternating Direction Method of Multipliers Based Distributionally-Robust Chance Constrained Optimal Power Flow," *Applied Energy*, vol. 239, pp. 938–947, 2019.

[145] R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications," *IEEE Transactions on Circuits and Systems*, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[146]J. H. Lee, "A 1.5-V 10-bit 100-MS/s CMOS ADC," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 1811–1819, Dec. 2001.

[147] A. M. Niknejad and R. G. Meyer, *CMOS RFIC Design: A Low-Voltage, Low-Power Approach*. Boston, MA, USA: Springer, 2003.

[148] R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[149] S. K. Gupta, "Design of a low-power 10-bit CMOS ADC," *IEEE Trans. Circuits Syst. I: Regul. Pap.*, vol. 49, no. 6, pp. 845–853, Jun. 2002.

[150] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[151] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2003.

[152] M. Pedram and J. M. Rabaey, *Power Aware Design Methodologies*. Boston, MA, USA: Springer, 2002.

[153] A. Wang and A. P. Chandrakasan, "A 180 mV subthreshold FFT processor using a minimum energy design methodology," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.

[154] A. V. G. Shenoy, R. E. Barnett, and B. H. Calhoun, "Low-power multiplier design using asynchronous adiabatic logic," in *Proc. IEEE Int. Symp. Low Power Electron. Design*, 2005, pp. 126–131.

[155] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in *Proc. ACM/IEEE Design Autom. Conf.*, 2004, pp. 868–873.

[156] A. Bellaouar and M. I. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*. Boston, MA, USA: Kluwer, 1995.

[157] S. Mutoh et al., "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, Aug. 1995.

[158] J. S. Wang and C. T. Hsieh, "High-speed and low-power CMOS full adder cells," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1999, pp. 187–190.

[159] S. M. Sze, *Physics of Semiconductor Devices*, 2nd ed. New York, NY, USA: Wiley, 1981.

[160] A. M. Niknejad and R. G. Meyer, *CMOS RFIC Design: A Low-Voltage, Low-Power Approach*. Boston, MA, USA: Springer, 2003.

[161] B. Razavi, *RF Microelectronics*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2011.

[162] M. H. Perrott, "A 10-bit 100-MS/s CMOS ADC with a 3.5-bit ENOB at Nyquist," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1699–1708, Dec. 2002.

[163] A. M. Niknejad and R. G. Meyer, "A 1.5-V 10-bit 100-MS/s CMOS ADC," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 1811–1819, Dec. 2001.

[164] S. K. Gupta, "Design of a low-power 10-bit CMOS ADC," *IEEE Trans. Circuits Syst. I: Regul. Pap.*, vol. 49, no. 6, pp. 845–853, Jun. 2002.

[165] J. H. Lee, "A 1.5-V 10-bit 100-MS/s CMOS ADC," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 1811–1819, Dec. 2001.

[166] R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[167] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[168] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2003.

[169] M. Pedram and J. M. Rabaey, *Power Aware Design Methodologies*. Boston, MA, USA: Springer, 2002.

[170] A. Wang and A. P. Chandrakasan, "A 180 mV subthreshold FFT processor using a minimum energy design methodology," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.

[171] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[172] A. V. G. Shenoy, R. E. Barnett, and B. H. Calhoun, "Low-power multiplier design using asynchronous adiabatic logic," in *Proc. IEEE Int. Symp. Low Power Electron. Design*, 2005, pp. 126–131.

[173] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in *Proc. ACM/IEEE Design Autom. Conf.*, 2004, pp. 868–873.

[174] A. Bellaouar and M. I. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*. Boston, MA, USA: Kluwer, 1995.

[175] S. Mutoh et al., "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, Aug. 1995.

[176] J. S. Wang and C. T. Hsieh, "High-speed and low-power CMOS full adder cells," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1999, pp. 187–190.

[177] L. Wei, Z. Chen, K. Roy, and V. De, "Design and optimization of dual-threshold circuits for low-voltage low-power applications," *IEEE Trans. VLSI Syst.*, vol. 7, no. 1, pp. 16–24, Mar. 1999.

[180] B. H. Calhoun and A. Chandrakasan, "Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 238–245, Jan. 2006.

List of Publications

S.No.	Title of paper with author names	Name of journal / conference	Published date	Issn no/ vol no, issue no	Indexing in Scopus/ Web of Science/UGC-CARE list (*please mention*)
1.	REVIEW OF EMERGING TRENDS IN LOW POWER MULTIPLIER Sheetal Nagar , Govind Singh Patel , Seema Nayak	Scopus Index International journal,PJAEE	02-Dec-20	Volume 17, Issue 9	Scopus
2.	Design of low power Digital Multiplier Sheetal Nagar , Govind Singh Patel , Seema Nayak	JETIR .	07-Nov-20	Volume 6, Issue 1	UGC care Listed
3.	A COMPARATIVE STUDY ON APPROXIMATE LOW POWER MULTIPLER Sheetal Nagar , Govind Singh Patel , Seema Nayak	International Conference ICRACI4.0	22-Dec-22	Volume 65, Issue 4	Scopus Index Journal, Springer and IOP Publishing

4.	A Design and Investigation Inexact compressor based on low power Multiplier Sheetal Nagar,Shanky Saxena, Govind Singh Patel, SeemaNayak	Journal of circuits, Systems and Computers	17-Aug-23	Volume 32, Issue 17	SCI
5.	Designing of Parameter of Exact and Inexact compressor based low power Multiplier Sheetal Nagar,Shanky Saxena, Govind Singh Patel,Seema Nayak	International Conference AECE-2023	15-Feb-2024	Volume 32, Issue 17	IEEE Explorar
6.	Verification, Analysis, validation and Implementation on FPGA of Inexact compressor based on low power Multiplier Sheetal Nagar, Shanky Saxena, Govind Singh Patel,Seema Nayak	International Conference at LPU ICICS-2024	Presented	------	Scopus Index Journal, Springer and IOP Publishing