

DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE MULTIPLIER USING 15:4 COMPRESSOR

A Main Project Report submitted in partial fulfillment of the

requirements for the award of degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted

By

P.INDU 22A95A0415

S.ALEKHYA 21A91A04G6

SK.MUSTAK 21A91A04B0

S.NIKITHA 21A91A04B2

Under the esteemed guidance of

Mr. S. HARICHANDRA PRASAD, MTech, (PhD)

Associate Professor



**ADITYA
UNIVERSITY**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

ADITYA UNIVERSITY

(Formerly Aditya Engineering College (A))

2024-2025

ADITYA UNIVERSITY

(Formerly Aditya Engineering College(A))

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



CERTIFICATE

This is to certify that the project work entitled "**DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE MULTIPLIER USING 15:4 COMPRESSOR**" is being submitted by

P.INDU	22A95A0415
S.ALEKHYA	21A91A04G6
SK.MUSTAK	21A91A04B0
S.NIKITHA	21A91A04B2

in partial fulfillment of the requirements for award of the B.Tech degree in Electronics & Communication Engineering.

Project Guide

Mr. S. Harichandra Prasad (Ph.D),
Asst. Professor

Head of the Department

Dr. Sanjeev Kumar, Ph.D.
Assoc. Prof., Dept. of ECE

External Examiner

DECLARATION

I hereby declare that the project entitled “**DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE MULTIPLIER USING 15:4 COMPRESSOR**” is a genuine project. This work has been submitted to the **ADITYA UNIVERSITY (Formerly Aditya Engineering College (A))**, Surampalem in partial fulfillment of the **B.Tech.**, degree . I further declare that this project work has not been submitted in full or part of the award of any degree of this or any other educational institutions.

by

P.INDU	22A95A0415
S.ALEKHYA	21A91A04G6
SK.MUSTAK	21A91A04B0
S.NIKITHA	21A91A04B2

ACKNOWLEDGEMENT

It is with immense pleasure that I would like to express my in debuted gratitude to my project Guide **Mr. S. Harichandra Prasad, Asst Professor.** who has guided me a lot and encouraged in every step of the project work. his valuable moral support and guidance throughout the project helped me a greater extent.

I owe our sincere gratitude to my project coordinator, **Dr. A L Siridhara, Assoc. Professor** for providing a great support and guidance throughout the project.

I am grateful to **Dr. Sanjeev Kumar, Assoc. Prof. and HOD,** for inspiring us all the way and for arranging all the facilities and resources needed for my project.

I wish to thank our **Dr. M.V. Rajesh, Assoc. Dean and Dr. Dola Sanjay S, Dean School of Engineering** for their encouragement and support during the course of my project.

I would like to extend my sincere thanks to **Dr. G. Suresh, Registrar, Dr. S. Rama Sree, Pro Vice-Chancellor, Dr. M.B. Srinivas, Vice-Chancellor, Dr. M. Sreenivasa Reddy, Deputy Pro Chancellor and Management, Aditya University** for unconditional support for providing me the best infrastructural facilities and state of the art laboratories during my project.

Not to forget, **Faculty, LabTechnicians, Non-Teaching Staff and our Friends** who have directly or indirectly supported me in completing this project on time



(Formerly Aditya Engineering College (A))

VISION & MISSION OF THE UNIVERSITY

VISION

Aditya University aspires to be a globally recognised academic institution dedicated to quality education, cutting-edge research, and technological service to our country, and envisions itself as a beacon of holistic advancement and long-term impact, remaining dynamic in the ever-changing worlds of society, ecology, and economics.

MISSION

M1: Aditya University pushes boundaries to design high-quality curricula and to provide students with a vibrant and relevant education that prepares them for a changing world. Our industry insights and creative teaching methods attempt to equip our students to be lifelong learners.

M2: Aditya University's learning environment encourages intellectual curiosity, critical thinking, and cooperation, with the goal of providing students with an immersive education that fosters creativity and innovation. Our cutting-edge facilities, interactive classrooms, and supportive faculty aim to motivate students to realise their full potential and contribute to society.

M3: Aditya University promotes cross-disciplinary inquiry and discovery and leads cutting-edge research and innovation. Through strategic partnerships, research grants, and a dedicated faculty, we aim to advance science, technology, and social sciences and empower students and faculty to conduct transformative research that solves real-world problems and elevates our institution globally.

M4: Aditya University is committed to producing world-changing business leaders and entrepreneurs through its emphasis on entrepreneurship, mentoring, and business incubation programmes.



ADITYA UNIVERSITY

(Formerly Aditya Engineering College (A))

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VISION & MISSION OF THE DEPARTMENT

VISION

To become a center of excellence in the field of Electronics and Communication Engineering with technological capability, professional commitment and social responsibility.

MISSION

- M1. Provide quality education, well-equipped laboratory facilities and industry collaboration.
- M2. Promote cutting edge technologies to serve the needs of the society and industry through innovative research.
- M3. Inculcate professional ethics and personality development skills.

Head of the Department

Head of the Department
Dept. of Electronics & Communication Engg
ADITYA UNIVERSITY



ADITYA UNIVERSITY

(Formerly Aditya Engineering College (A))

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Graduates of the Program will

PEO 1:

Adapt the learning culture needed for a successful professional career and pursue research.

PEO 2:

Build modern electronic systems by considering technical, environmental and social contexts.

PEO 3:

Communicate effectively and demonstrate leadership qualities with professional ethics.

Head of the Department

Head of the Department
Dept. of Electronics & Communication Engg
ADITYA UNIVERSITY



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROGRAM OUTCOMES (POs)

After successful completion of the program, the graduates will be able to

PO 1	Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
PO 2	Identify, formulate, research literature and analyze complex engineering problems, reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
PO 3	Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO 4	Conduct investigations of complex problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.
PO 5	Create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering activities, with an understanding of the limitations.
PO 6	Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.
PO 7	Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of, and need for sustainable development.
PO 8	Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
PO 9	Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.
PO 10	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	Demonstrate knowledge and understanding of engineering management principles and apply these to one's own work, as a member and leader in a team and to manage projects in multidisciplinary environments.
PO 12	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



Head of the Department

Head of the Department
Dept. of Electronics & Communication Engg
ADITYA UNIVERSITY



ADITYA UNIVERSITY

(Formerly Aditya Engineering College (A))

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROGRAM SPECIFIC OUTCOMES (PSOs)

After successful completion of the program, the graduates will be able to

PSO 1:

Provide sustainable solutions in the field of Communication and Signal Processing.

PSO 2 :

Apply current technologies in the field of VLSI and embedded systems for professional growth.

Head of the Department

Head of the Department
Dept. of Electronics & Communication Engg
ADITYA UNIVERSITY



Course Outcomes

After completion of the course the graduates will able to attain the following course outcomes.

CO1: Formulate a real world engineering problem through thorough investigation.

CO2: Design the methodology for project work plan, schedule, and cost.

CO3: Apply the domain knowledge and modern tools to arrive at a framework to solve the problem.

CO4: Analyze the obtained solution within the context of an engineering framework that addresses societal and environmental concerns while adhering to professional ethics.

CO5: Prepare a technical report with effective written communication skills.

CO6: Interpret the results of project work with oral communication skills.

A handwritten signature in black ink, appearing to read "S. S. S." followed by a horizontal line.

Head of the Department

Head of the Department
Dept. of Electronics & Communication Engg
ADITYA UNIVERSITY



(Formerly Aditya Engineering College (A))

Project-PO Mapping

A.Y: 2024-25 (AR 20)

Sem.: VIII

Section & Batch: C1

Project Title: **DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE**

MULTIPLIER USING 15:4 COMPRESSOR

Abstract

The design and parametric analysis of a flower-shaped patch antenna. The proposed patch antenna is designed using FR-4 substrate mounted on the Jerusalem cross-shaped DGS structure. A Double Negative (DNG) triple Complementary Split Ring Resonator is embedded inside the substrate. A circular foam substrate with the dimensions of $10 \times 4 \times 10 \times 2$ mm³ is sandwiched between the patch and the FR-4 substrate. The overall dimensions of the patch are 23.5×16 mm. The proposed antenna resonates at 5.2 GHz and 8.25 GHz respectively. This also possesses wide bandwidth of 1.2 GHz (24.1%) in the range of 4.95–6.15 GHz and 2.2 GHz (26.5%) in the range of 7.1–9.3 GHz. The gains in these bands are observed to be 3.93 db and 5.02 db respectively. The whole design is carried out in High Frequency Simulation Structure Integration Technique. The developed multiband antenna can be useful for several wireless communication applications, such as WLAN, Wi-MAX, and ISM band. The proposed antenna is fabricated and its performance parameters are measured. The simulated and measured results are in good agreement.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
OverallMapping	3	2	3	2	3	2	1	1	3	2	1	2

	PSO1	PSO2
OverallMapping	3	1

Signature of Project Members:

1.20A91A0413

2.20A91A0414

3.20A91A0415

4.20A91A0416

Guide Signature

Contents

	Page No.
List of Figures	I
List of Tables	II
Nomenclature	III
Abstract	IV
1. INTRODUCTION	4-11
1.1 INTRODUCTION TO APPROXIMATE ARITHMETIC UNIT	4-5
1.2 TYPES OF MULTIPLIERS IN VLSI	5-7
1.2.1 Array Multiplier	6
1.2.2 Wallace Tree Multiplier	6
1.2.3 Dadda Multiplier	7
1.2.4 Booth Multiplier	7
1.2.5 Sequential Multiplier	7-8
1.3 ROLE OF APPROXIMATE COMPUTING	8-9
1.4 OBJECTIVES OF THE PROJECT	9-10
1.5 ORGANIZATION OF THE THESIS	10-11
2. LITERATURE SURVEY	12-21
2.1 ADDER	12-15
2.2 COMPRESSORS	15-18
2.3 MULTIPLIERS	19-21
3. DESIGN OF APPROXIMATE MULTIPLIER	22-26
3.1 RELATED APPROXIMATE COMPRESSORS	22-23
3.1.1 5:2 Compressor	23-24
3.1.2 6:2 Compressor	24
3.1.3 7:2 Compressor	25
3.1.4 7:3 Compressor	26

4. INTRODUCTION TO TECHNOLOGY

4.1 VLSI

4.2 About Verilog HDL

4.3 Tools

 4.3.1 Xilinx

 4.3.2 Introduction to Xilinx ISE

 4.3.3 Xilinx software

 4.3.3.1 Creating a new project

5. EXISTING METHOD

5.1 Block diagram of Existing method

5.2 Existing Compressor and Adder

 5.2.1 3:2 compressor

 5.2.2 4:2 compressor

 5.2.3 Han-carlson adder

6. PROPOSED METHOD

6.1 Block diagram of Proposed method

6.2 Proposed compressor and Adder

 6.2.1 5:3 Compressor

 6.2.2 15:4 Compressor

 6.2.3 Kogge stone adder

7. Result and discussion

7.1 Code for 16-bit proposed multiplier

7.2 Implementation

7.3 Comparision

7.4 Conclusion

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1.	Different stages of parallel-prefix adder	13
2.	Methodology for approximate multiplier	22
3.	Diagram of 5:2 compressor	23
4.	Diagram of 6:2 compressor	24
5.	Diagram of 7:2 compressor	25
6.	Diagram of 7:3 compressor	26
7.	Block Digram of Han-Carlson Adder	42
8.	Block Digram Of 3:2 Compressor	43
9.	Block Digram Of 4:2 Compressor	45
10.	Han-Carlson Adder	47
11.	4-Bit Ripple Carry Adder	49
12.	4-Bit Carry Look ahead Adder(CLA)	50
13.	Carry select adder(CSL)	51
14.	Carry Save Adder(CSA)	52

15.	Structure of 16-bit wallet tree multiplier using 15:4 compressor	55
16.	Block representation of 5-3 compressor	56
17.	Block representation of approximate 5-4 compressor using modified approximate Full adder	58
18.	Logic Structure of the modified 3 bit approximate full adder	59
19.	Function of basic KSA	60
20.	Block representation of KSA	61
21.	Simulation result 16 bit multiplier	64
22.	Schematic diagram of 15-4 compressor	64
23.	Delay report of 16 bit multiplier	64
24.	Power report for 16 bit multiplier	65

List Of Tables

Table No.	Table Name	Page Number
1	Comparision of existing and proposed methods	66
2		

Nomenclature

1. Product Generation -PPG
2. Partial Product Reduction -PPR
3. Wallace Tree-WT
4. VLSI -very large scale integration
5. DSP- Digital Signal Processing
6. Register Transfer Level -RTL
7. CMOS- Complementary Metal-Oxide-Semiconductor
8. ALU- Arithmetic Logic Unit
9. parallel-prefix -PPx
10. Kogge-Stone (KSA)
11. Brent-Kung (BKA)
12. Han-Carlson (HCA)
13. FPGA- Field Programmable Gate Arrays
14. ASIC- Application-Specific Integrated Circuit
15. TSMC- Taiwan Semiconductor Manufacturing Company
16. Pass Transistor Logic -PTL
17. PMOS- P-channel Metal-Oxide-Semiconductor
18. 1NMOS- N-channel Metal-Oxide-Semiconductor
19. Digital Image Processing (DIP).
20. SMIC- Semiconductor Manufacturing International Corporation,
21. PSNR- Peak Signal-to-Noise Ratio
22. SSIM- structural similarity index measure
23. CSLA - Carry Select Adder
24. MBE- Molecular Beam Epitaxy
25. PDP- Power-Delay Product,
26. Chip-Area Ratio -CAR
27. error rate -ER
28. Error Distance -ED

- ,29.NED- Netlist Editor
- 30.integrated circuits (ICs)
- 31.Small-Scale Integration (SSI)
- 32.Medium-Scale Integration (MSI),
- 33.Large-Scale Integration (LSI)
- 34.Ultra-Large-Scale Integration (ULSI)
- 35.Deep Submicron (DSM)
- 36.GPU- graphics processing unit
- 37.FPGAs (Field Programmable Gate Arrays),
- 38.CPLDs (Complex Programmable Logic Devices),
- 39.SoCs (System on Chips).
- 40.HDL Design (VHDL)
- 41.Ripple Carry Adder (RCA)
- 42.Carry Look-Ahead Adder (CLA).
- 43. MSE- Mean Square Error

Abstract

The most widely used arithmetic operation is multiplication and it plays a crucial section in many applications. The concept of approximate computing has recently emerged as a promising design approach. The approximate computing technique is the key to lowering hardware complexity and improving energy, efficiency and performance. So, in this we will discuss about the modified approximated multipliers. The proposed algorithm will be better than existing in terms of area, hardware and power consumption by using concept of rounding and shifting. These multipliers are implemented in Verilog HDL using Xilinx.

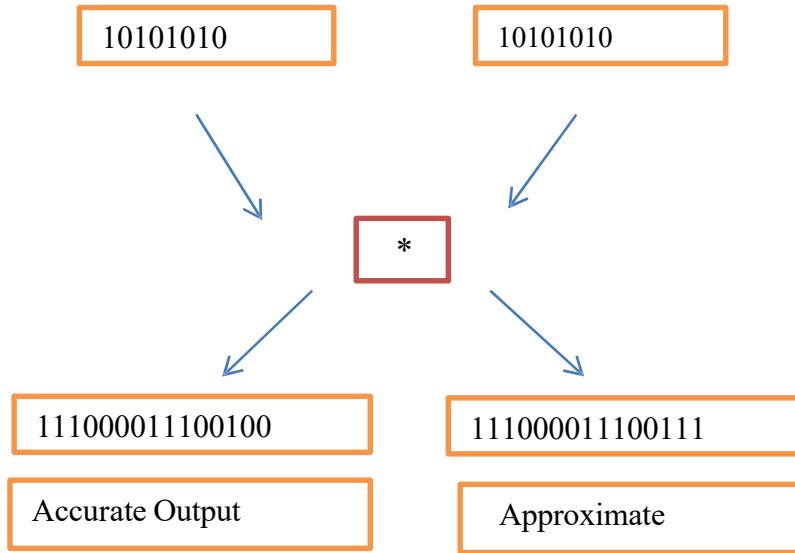
CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION TO APPROXIMATE ARITHMETIC UNIT

Arithmetic and Logic Units are the essential components of any digital Very Large-Scale Integration (VLSI) system. Realizing efficient ALU is required for better performance of a data path unit in microprocessors or Digital Signal Processors (DSP). Data path element actually perform computational operations like read/write to memory, arithmetic, logic operations, and numerical shift operations with elements like adder, subtractor, logical, and the shifting units etc. All microprocessors contain these elements in some form or another, satisfying price and performance constraints. The adders and multipliers are also essential for digital operations such as data conversions, filtering and convolutions in digital signal processing etc. The ability of data generations decides the speed and operations of every device. Among these multipliers are the significant elements that contribute for the total delay and hardware complexity in CMOS logic design. Hence this research concentrates on the design of multipliers suitable for data path systems.

The miniaturization of digital circuits generally can be achieved by either reducing the size of transistor or by optimizing the gate count of the circuit. The first approach extensively concentrates on transistor level designs. The latter approach deals with the application of Boolean rules and it is accomplished by logical analysis of digital design. The ultimate outcome is to have an area, power, delay optimized architecture/approach that optimizes the circuit at gate level. Digital arithmetic in logic design develops appropriate algorithms in order to achieve an efficient utilization of the available hardware. Since ultimately, speed, power and chip area are the most often used measures of the efficiency of an algorithm in hardware aspects, there is a strong link between the algorithms and technology used for its implementation. VLSI system in digital signal processing, image processing adds features through different types of algorithms for implementing multipliers and adders to give better results in processors design.



1.2 TYPES OF MULTIPLIERS IN VLSI

In digital system the multipliers are the most energy hungry blocks as they are fundamental sub-system for digital signal processors; application based embedded systems etc. Existing literatures introduce diverse multiplication methods by providing novelty in Partial Product Generation (PPG), Partial Product Reduction (PPR) and final addition to provide better performance in terms of area, power and delay metrics. Generally for the multiplication of an n -bit multiplicand with an m bit multiplier, m partial products are generated and product formed is $n + m$ bits long. Recent growths in processor design focus on low power multiplier architecture in most cases. In VLSI design the main concentration is done to get higher speed, lower cost, and less area. Taking into consideration these constraints, the design of low power multiplier is of great interest. Many research efforts in the multiplier design have been introduced to obtain energy efficiency in VLSI circuits. The various multiplication algorithms normally used are discussed in the following subsections.

1.2.1 Array Multiplier

Array multiplier has regular and simple structure. This multiplier uses repetitive addition and shifting to produce final product. Each partial product is generated by the multiplication of the multiplicand with one multiplier digit. The partial products are shifted according to their bit sequences and then added. The summation can be performed with normal carry propagation adder. $N-1$ adders are required where N is the no. of multiplier bits. Though the design time is less the array multiplier has disadvantages like high power consumption and requires digital gates resulting in large chip area in spite of its regular structure

1.2.2 Wallace Tree Multiplier

A fast multiplication of two numbers was developed by Australian computer scientist Chris Wallace (1964). This method witnessed that it is possible to perform the addition operations in parallel, resulting in less delay for multiplication. Parallel addition of the partial product bits using a tree of carry save adders, also known as —Wallace Tree (WT). A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers where partial product matrix is reduced to a two-row matrix by using a carry save adder and the remaining two rows are summed using a fast carry propagate adder to form the product. This is advantage for multipliers more than 16 bits. WT increases speed because the addition of partial products is now. In WT architecture, all the bits of all of the partial products in each column are added together by a set of counters in parallel without propagating any carries. Another set of counters then reduces this new matrix and so on, until a two-row matrix is generated. Wallace method uses three-steps to process the multiplication operation.

- Formation of bit partial products.
- The bit product matrix is reduced to a 2-row matrix by using a carry- save adder.
- The remaining two rows are summed using a fast carry- propagate adder to produce the product

1.2.2.1 Dadda Multiplier

A hardware multiplier design invented by computer scientist Luigi Dadda(1965) . It is similar to the Wallace multiplier, but it is slightly faster and requires fewer gates. Dadda and Wallace multipliers have the same three steps for multiplication. Wallace tree multipliers reduce as much as possible on each layer but Dadda multipliers reduce the number of gates used and delay.

1.2.3 Booth Multiplier

Booth multiplication algorithm gives a procedure for multiplying binary integers in signed -2's complement representation. An increasing number of high speed DSP applications have need of high precision fixed or floating point multiplier suitable for VLSI implementation. In Modified Booth multiplier partial products are added one at time using adder array whose final results is obtained with a final carry propagate adder stage. Modified booth algorithm reduces the number of partial products to be generated and is known as fastest multiplication algorithm.. Modified Booth multiplier performs both signed and unsigned multiplications.

1.2.4 Sequential Multiplier

To multiply two binary numbers (multiplicand X has n bits and multiplier Y has m bits) using single n bit adder, a sequential circuit that processes a single partial product at a time and then cycle the circuit m times can be used. Sequential multipliers are attractive for their low area requirement. In a sequential multiplier, the multiplication process is divided into some sequential steps. In each step sequence of partial products will be generated, added to the accumulated sum in the previous step, and the partial sum will be shifted to align the accumulated sum with partial product of nextstep. Therefore, each step in sequential multiplication consists of three different operations viz., generating partial products, adding the generated partial products to the accumulated partial sum, and shifting the partial sum. Figure 1 shows partial product generation and addition in a sequential multiplier.Todays embedded applications, mobile systems etc., depends on reduction of energy consumption. Many dedicated systems are developed to get

better energy efficiency at different levels like software level, architecture level, circuit, device levels etc. Methods that deal energy at the algorithmic level use exact building blocks. But in real time many application can tolerate limited and small errors to form inexact but low-power building blocks. Low power in the field of VLSI can be used in handy multimedia devices where useful information from a little erroneous output employing various signal processing algorithms and architectures is accepted. Therefore for such applications there is no need to produce exactly correct numerical outputs. So the proposed research focuses on design of efficient architectures using approximate computing and compares the efficiency with accurate counterparts.

1.3 ROLE OF APPROXIMATE COMPUTING

Many scientific and engineering problems are computed using accurate, precise and deterministic algorithms. However, in many applications involving signal/image processing and multimedia, exact and accurate computations are not necessary, because these applications are error tolerant and produce results that are good enough for human perception. In these error resilient applications, a reduction in circuit complexity, and thus area, power and delay is very important for the operation of a circuit. Hence approximate computing can be used in error tolerant applications by reducing accuracy, but still providing meaningful results faster and with lower power consumption. Many domains, like multimedia and big data analysis, exhibit an intrinsic acceptance to a certain level of inaccuracies in computation. Functional approximation, in hardware, mostly deals with the design of approximate arithmetic units, such as adders and multipliers, at different abstraction levels, i.e., transistor, gate, Register Transfer Level (RTL) and application. Approximate techniques are needed to implement algorithm in image processing and video processing achieving persistent, portable nature of electronic gadgets to tackle the present scenario of increasing demand for ultra-low power consumption, small area, and high performance objectives by compromising the accuracy.

Energy-efficient design techniques are addressed in many literatures at all levels of design hierarchy. Schemes at lower level design process such as logic and circuit levels are typically application independent. At the algorithmic and architectural levels, features are precise to a category of applications and these lead to the development of application specific energy

reduction techniques. The scaling of CMOS technology plays a major role in reducing energy consumption of circuit. The reduction in energy dissipation is also possible via voltage scaling. The amount of voltage scaling is limited by the critical path delay of the architecture and the throughput requirements of the application. Reduction in power consumption of the chip is achieved, when both the CMOS technology and the design complexity are scaled in parallel to obtain the required functionality. Transistor count is a primary concern which largely affects the design complexity of many functional units such as Multiplier and ALU. Moore's law states the transistors need for VLSI design and gives the experimental observation of component density and performance of integrated circuits, which doubles every two years.

1.4 Objective of the Paper

The primary objective of this paper is to design and analyze a 16×16 -bit Wallace Tree Multiplier using approximate compressors, specifically focusing on the 15-4 compressor and integrating a Kogge Stone Adder for final-stage addition. The goal is to optimize power efficiency, area reduction, and speed enhancement while maintaining acceptable error rates for applications like multimedia and image processing. Key objectives include:

- Implementing Approximate Computing Techniques
- Reducing power dissipation and critical path delays by replacing exact compressors with approximate 5-3 compressors in the 15-4 compressor design.
- Evaluating error metrics such as Error Rate (ER), Error Distance (ED), and Normalized Error Distance (NED) to ensure reliability in error-resilient applications.
- Designing a High-Performance 15-4 Compressor

Structuring the 15-4 compressor using three phases:

Phase 1: Five full adders.

Phase 2: Two approximate 5-3 compressors.

Phase 3: A 4-bit Kogge Stone Adder for final carry propagation.

- Minimizing area and delay by leveraging parallel processing in the Kogge Stone Adder.

- Optimizing the 16×16 Wallace Tree Multiplier
- Integrating the proposed 15-4 compressors into the partial product reduction stage of the multiplier.
- Enhancing speed by adding "0" bits in higher-order columns (13th–15th) to align with the 15-4 compressor inputs.
- Performance Evaluation and Comparison
- Synthesizing and simulating the design using Xilinx ISE 14.5 to analyze area, power, and timing.

Comparing results with accurate multipliers and other approximate designs to validate improvements in:

- Device utilization (e.g., LUTs, slices, flip-flops).
- Power-delay product (PDP) and area-delay product (ADP).
- Future Scalability
- Proposing extensions for higher-order compressors (e.g., 31-5) to support larger multipliers (32×32 bits) while maintaining efficiency.
- This work bridges the gap between approximate computing and high-speed arithmetic architectures, offering a balanced solution for resource-constrained, error-tolerant systems.

1.5 Organization of the Thesis:

Chapter 1: Introduction

- Introduces approximate arithmetic units and their relevance in VLSI systems.
- Discusses the importance of multipliers in digital signal processing and the need for energy-efficient designs.
- Defines the objectives of the research, focusing on designing area-efficient, low-error, and optimized approximate multipliers for image and signal processing applications.

Chapter 2: Literature Review

- Reviews existing research on arithmetic circuits for approximate computing.
- Explores their applications in digital image and signal processing.

Chapter 3: Design of Approximate Multiplication Algorithm

- Describes the development of an approximate multiplication algorithm using 4:2 approximate compressors.
- Details the mathematical foundation and corresponding VLSI architecture.
- Includes performance analysis comparing area, power, delay, PDP, and ADP metrics with conventional designs.

Chapter 4: Error-Efficient Precise Approximate Multiplier Design

- Explains the design of $n \times nn \times n$ error-efficient precise approximate multipliers using Dadda structure-based PP arrangement.
- Utilizes approximate compressors in least significant columns and exact compressors in most significant columns.
- Synthesized using ASIC platform with 90 nm PDK technology.

Chapter 5: Area-Efficient Precise Multiplier Design

- Proposes three variants of area-efficient precise multipliers for portable systems.
- Introduces approximate compressors that generate no error in sum signals to minimize errors.
- Synthesized using structural Verilog HDL in ASIC platform with 90 nm PDK technology.

Chapter 6: Implementation in Digital Image and Signal Processing Applications

- Implements proposed multipliers in image enhancement systems (e.g., smoothing and scaling architectures).
- Prototypes hardware on Spartan 6 FPGA using Xilinx-MATLAB co-simulation with System Generator tool.
- Evaluates visual quality, PSNR, and MSE metrics by comparing output images processed by proposed designs with prior approximate designs.

Chapter 7: Summary and Future Work

- Summarizes the approximate multiplication algorithms proposed in the thesis.
- Provides recommendations for future research directions.

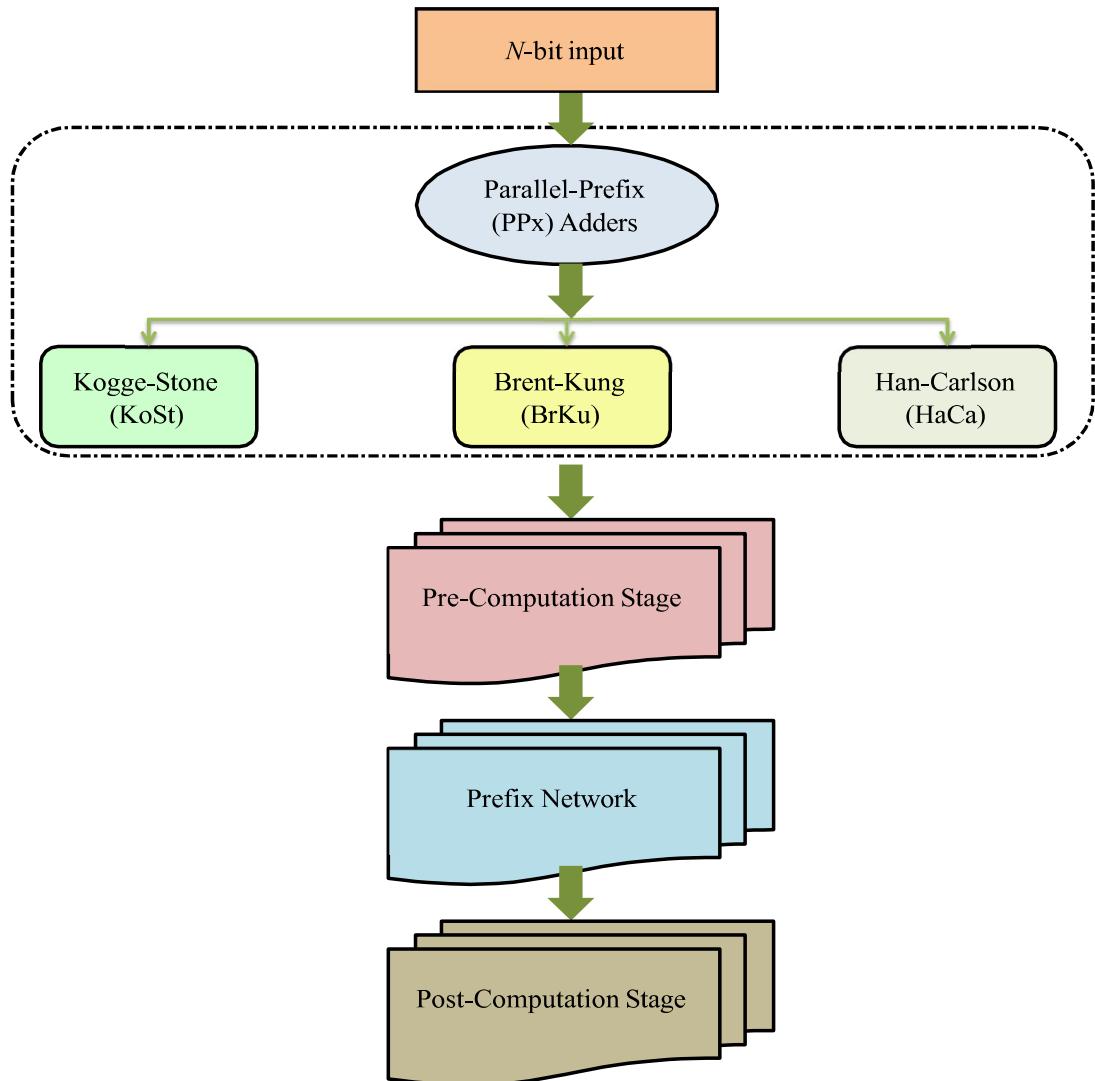
CHAPTER 2

LITERATURE SURVEY

2.1 ADDER

Addition is a fundamental function in arithmetic operation; several adder designs have been proposed for implementations in inexact computing. These adders show different operational profiles; some of them are approximate in nature while others rely on probabilistic features of nanoscale circuits. However, there has been a lack of appropriate metrics to evaluate the efficacy of various inexact designs.

In this research work, parallel-prefix (PPx) adders are considered in comparison to serial adders. The structure of serial adders is less complex but dissipates more power. The addition in serial adders is executed sequentially, this is a time-consuming process. To overcome this problem PPx adders are considered. In PPx adders, prefix operation is done for efficient addition. PPx adders offer a solid theoretical foundation for a variety of design trade-offs in terms of power consumption, speed, and area utilization. Adders are a common circuit, and their regularity makes them an ideal choice for VLSI synthesis, which may also be used to evaluate design trade-offs. Figure 2.1 shows the flow diagram for computation stages of parallel-prefix adders.

**Figure 2.1.1:** Different stages of Parallel-Prefix adder

Andrew Kahng & Seokhyeong Kang (2012) proposed an area- error optimized approximate adders in which accuracy can be configured during run time. Evaluations revealed that area efficient variant demonstrated better power and delay reductions at the expense of sacrifice in accuracy while the accuracy efficient variant demonstrated better error reduction at the expense of high delay and power dissipation.

Jinghang Liang et al. (2013) analysed the trade-off between performance and error metrics of approximate adders that are reconfigurable between exact and inexact modes. The error metrics considered are viz., Error distance, normalized error distance, mean error distance,

while the performance metrics considered are viz., area, power, and delay.

Vaibhav Gupta *et al.* (2013) proposed a multibit adder targeting digital signal processing applications in Discrete cosine transform and Finite Impulse Response (FIR) filter. The proposed multi-bit adder incorporates various transistor designs of approximate adder cells. Approximation is done in the full adder cells with logic level changes at transistor level. Implementation results demonstrated significant area, and power reductions of proposed adder based DSP systems compared to the standard adder based DSP systems.

Muhammad Shafique *et al.* (2015) proposed a reconfigurable adder that can work in variable approximation modes. The design incorporates an error correction circuitry and it can be configured to vary the accuracy level of the proposed adder. Experimental results demonstrated that the proposed reconfigurable adder exhibits lower latency when compared to the state-of-the-art approximate adders. Functionality of the proposed reconfigurable adder is verified with prototype on Xilinx Virtex-6 FPGA board.

Pawan Sonwane *et al.* (2015) proposed the design of low power inexact 4:2 compressor using approximation algorithm. The proposed inexact 4:2 compressor trades-off accuracy at the benefit of power reduction. Analysis revealed that the proposed design fair better when compared to prior designs.

Jothin & Vasanthanayaki(2018) proposed a High Performance Error Tolerant Adders (HPETA) using Multiplexer Based Approximate Full Adder cells in the inaccurate part. Simulations using Cadence Encounter with TSMC 90-nm ASIC technology revealed that the proposed adder design exhibit high speed, low energy, and less Area-Delay Product when compared to the recent previous approach.

Ayad Dalloo *et al.* (2018) proposed a novel architecture for approximate adder incorporating OR gates in the least significant part instead of full adder cells. The OR gates in the least significant part not only reduces the gate count but produces better accuracy compared to the other approximate logic based adder cells. Implementation of the proposed methodology in an 8-bit approximate adder achieves improvement in the mean squared error by 58.5%, compared to the previously reported best architecture.

Gnanambikai Palanisamy *et al.* (2019) proposed an error-tolerant parallel adder with faithful approximation that can optimise area and accuracy. The proposed adder performs parallel operation using carry select algorithm and utilizes two variants of approximate full

adder cells in the least significant part. Evaluations of the proposed approximate full adder cells are carried out by comparing with exact full adder cell implemented adder. Evaluations revealed that exact full adder cell implemented version perform better in accuracy while the approximate adder cells implemented versions perform better in terms of power and area reductions. Driving capability and functionality of the proposed parallel adders are verified with implementations in digital image and digital signal processing applications.

2.2 COMPRESSORS

Hsiao *et al.* (1998) proposed a high-speed low-power full adder and 4–2 compressor targeting PP compression in multipliers. The proposed methodology concentrates on the reduction of capacitance in the data-path of the arithmetic elements. Performance evaluation in CMOS analog design environment revealed that the proposed arithmetic units outperform in terms of power and delay reductions in multiplier implementations compared to the prior multiplier designs.

Chang *et al.* (2003) proposed novel 4–2 and 5–2 compressors in CMOS technology that can operate at ultra-low supply voltage. In the proposed approach various architectures are proposed that incorporates either circuit level modifications or gate level modifications. Pass Transistor Logic (PTL) was used to design the proposed circuits to reduce area. However the weak driving capability of PTL was overcome with PMOS-NMOS feedback transistors. Evaluations reveal that the proposed arithmetic units are able to perform well in multiplier implementations in terms of power reduction and driving capability. In addition, simulation results show that Chang *et al.* (2003) 4-2 compressor with the proposed XOR–XNOR module is able to function at low supply voltage of 0.6 V, and outperform prior CMOS logic compressors proposed in literature.

Minho ha & Sunggu Lee (2018) proposed low-power error- efficient approximate multiplier targeting Digital Image Processing (DIP). The proposed multipliers use new propose 4:2 compressor in few least significant columns and exact 4:2 compressor in the most significant columns. However the error due to approximation in the least significant columns is reduced using error-recovery modules. Evaluations revealed that the proposed multiplier outperform similar designs in terms of power and error reductions.

Guo Y *et al.* (2018) proposed a low-error approximate multiplier by employing new novel approximate compressors for PP compression. The error percentage of the approximate multipliers are configured at run time using the proposed compressors. Evaluations in CMOS SMIC 40nm process technology revealed that the best accurate proposed multiplier demonstrate significant power, area, and delay reductions compared to the standard Wallace tree multiplier. In addition the functionality of the proposed approximate multipliers are verified with image processing application.

Yen-Jen Chang *et al.* (2019) proposed fault-tolerant 4-2 compressor targeting PP compression in the multipliers. The main significance of the proposed work is that the proposed 4:2 compressors are configured based on the pattern distribution of digital images considered in the application. The proposed compressors and targeted multipliers are designed using 90nm CMOS technology and simulations reveal that the proposed compressors and targeted multipliers outperform state-of-the-art approaches in terms of power and error reductions. Implementation of the proposed multipliers in digital image processing reveal superior output image quality with better PSNR and SSIM metrics.

Kenneth Steiglitz & Peter R. Cappello (1983) proposed a fast parallel counter targeting PP accumulation in Dadda multiplier. Simulation results revealed that the proposed multiplier is able to demonstrate high speed compared to prior designs.

Ming-Roun Jiang et al. (1998) proposed a novel high-speed and low power 3-2 counter and a 4-2 compressor for PP summation in array multiplier. Synthesis results demonstrated less internal load capacitance that contributed for higher speed and power performance than prior approaches.

Radhakrishnan & A.P. Preethy (2000) proposed a transistor level CMOS 4-2 compressor targeting multiplier implementation. PTL is used to reduce area of the design but at the expense of speed.

Jiangmin Gu & Chip-Hong Chang (2003) proposed a low power 4- 2 compressor at transistor level using low-power XOR-XNOR gate that is able to operate with ultra-low supply voltage. Simulation results show that the proposed 4-2 compressor demonstrate better performance compared to prior arts and is able to operate at supply voltage as low as 0.6V.

Dursun Baran *et al.* (2010) proposed energy efficient 3:2 and 4:2 compressors. The functionality of the proposed compressors are analysed with implementations in 16 bit Booth and Non-Booth multipliers. Synthesis results demonstrated that Non-Booth multipliers are more energy efficient compared to Booth counterparts.

Pourormazd M *et al.* (2011) proposed low power serial multiplier and serial adder, combinational booth multiplier and shift/add multipliers, targeting design of digital FIR filters. Synthesis results demonstrated that shift/add multiplier based FIR filter exhibit better performance in terms of power dissipation and high speed improvement.

Momeni & Lombardi (2015) proposed novel approximate 4-2 compressors targeting PP reduction in multiplier. Extensive simulations with implementations in Dadda multiplier revealed that the proposed designs perform better in image processing applications. In addition the performance metrics revealed that the proposed designs fair significant reductions in power dissipation, delay and transistor count compared to standard design.

Omid Akbari *et al.* (2017), proposed approximate 4:2 compressors, that are reconfigurable to work both in exact and approximate operating modes. In the approximate mode, these dual-quality compressors provide better speed improvement and dissipate less power

.

The proposed compressors are employed for PP reduction in multipliers. Evaluations with Dadda multiplier implementations with standard CMOS technology revealed that the proposed multipliers achieve lower delay and power consumption in the approximate mode. In addition implementations in image processing applications revealed better processed images with high SSIM metric

Yi Guo *et al.* (2018) proposed probability –driven inexact compressors and inexact half- adders targeting PP reduction in multipliers. Several error reducing schemes are proposed to keep the error in the multiplier output within a fixed range. Evaluations revealed that the proposed multipliers perform best in terms of error minimization compared to the best of prior approximate designs and demonstrate 50.52%, 52.46%, 33.90% reduction in power, area and delay, respectively, compared to the conventional counterpart

Minho ha & Sunggu Lee (2018) proposed approximate 4-2 compressors for PP accumulation in multiplier. Error recovery Modules are proposed and used to reduce error due to approximate compressors in PP columns. The error in the proposed multiplier is bound within range by use of approximate compressor in least significant PP columns and exact compressor in the most significant PP columns.

Karri Manikantta Reddy *et al.* (2019) proposed a novel approximate 4–2 compressor and utilized it for PP accumulation in Dadda Multiplier .The proposed multiplier is targeted for error resilient applications in digital image processing. Evaluations in a 45 nm standard CMOS technology revealed that the proposed compressor realize a significant reduction in error rate compared to the similar prior approximate designs , and proposed multiplier achieved 35%, 36% and 17% reduction in power consumption, delay and area respectively compared to the conventional counterpart. Implementations revealed that the proposed multiplier demonstrated nearly 90% structural similarity in processed outputs compared to the output images processed by conventional multiplier based systems

2.3 MULTIPLIERS

Zhongde Wang *et al.* (1995) proposed novel full and half adders for PP compression in multipliers. Evaluations revealed that the proposed multiplier fair better interms of area and node capacitance reductions. Implementations in higher end 2s complement multiplier revealed their functionality and driving capability.

Guoping Wang & James Shield (2005) proposed an area efficient array multiplier that exhibits better power and delay reductions. Evaluations demonstrated that the proposed multiplier exhibit better performance compared to the previous similar designs. Implementations revealed that the proposed scheme is more suitable for FPGA prototype development.

Chip-Hong Chang & Ravi Kumar Satzoda (2010) proposed a multiplexer based array multiplier utilizing adaptive pseudo-carry generation circuitry with least significant bit truncation. The adaptive pseudo-carry generation circuitry demonstrates low average error compared to prior truncation multipliers. Evaluations demonstrated that the proposed pseudo truncation array multiplier achieves 25% and 40% reductions in silicon area and dynamic power, respectively when compared to conventional full-width multiplier for 32-bits in input operand. Also the implementations in digital image processing systems revealed that the proposed multiplier is able to produce output images with SSIM metric close to the conventional multiplier based systems

Khaing Yin Kyaw *et al.* (2010) proposed a novel approach for multiplier that engaged power, delay and accuracy metrics in design parameter. Evaluations revealed that the proposed multiplier outperform peers in terms of power reduction and speed improvement

Chia-Hao Lin & Ing-Chao Lin (2013) proposed a inaccurate 4-2 counter and implemented for PP reduction in Wallace multiplier. To reduce error due to inaccurate compressors, Error detection and correction circuitry is incorporated in Wallace multiplier. Experimental results demonstrated that the proposed multiplier exhibit 10.74%, and 9.8% reductions in power consumption and of delay, respectively compared to the conventional counterpart

Cong Liu *et al.* (2014) proposed an error tolerant multiplier targeting few applications in DSP. The proposed multiplier incorporates an approximate adder for PP

Compression that reduces carry propagation delay encountered in conventional adders. Additionally the proposed multiplier utilizes a configurable error recovery module to reduce error due to approximate adders in PP columns. Simulations revealed that the proposed multiplier exhibit 20% and up to 69% reductions in delay and power, respectively compared to the standard Wallace multiplier.

Zervakis *et al.* (2015) proposed an area efficient approximate multiplier using heuristic optimization technique for design evaluation viz., synthesis, simulation, power and timing analysis. Experimental analysis revealed that the algorithm optimized multiplier design is able to perform better when compared to the state-of-art approaches. Evaluations reveal that the proposed multiplier is able to realize power savings of 11% to 30% for variation in error bounds from small to large.

Srinivasan Narayananamoorthy *et al.* (2015) proposed a novel multiplier architectures that can tradeoff computational accuracy with energy consumption. Simulations revealed that the proposed multiplier consumes 58% less energy compared to the standard multiplier with average computational error of not more than 1%. Implementations in DSP applications revealed that the proposed multiplier perform close to standard design and the error exhibited is noticeably tolerable.

Manjunath *et al.* (2015) proposed the design of 16*16 Modified Booth multiplier with Carry Select Addition algorithm for PP accumulation. Evaluations revealed that the proposed CSLA based MBE multiplier demonstrate minimum hardware, and low power dissipation compared to the prior arts .

Suganthi Venkatachalam & Seok-Bum Ko (2017) Proposed an area efficient multiplier using approximate compressor units for PP accumulation. Optimizations in area and error metrics are achieved by varying logic depth in approximate compressor based on probabilistic models. Synthesis results of area efficient and error efficient variants of proposed multipliers revealed power savings of 72% and 38%, respectively, compared to the standard design. On an average, the proposed area and error efficient variants exhibited mean relative error of 7.6% and 0.02%, respectively, and it is significantly low compared to the prior approaches. Finally implementations in digital image processing application, of the proposed models achieved lowest MSE compared to other approximate designs.

Ihsen Alouani *et al.* (2018) proposed a new architecture for parallel multiplication that engages variants of approximate compressors for PP accumulation. Design evaluation revealed that the multi variant approximate compressor based parallel multiplier achieves better performance in power, area and delay metrics coupled with error minimization when compared to the single approximate compressor based multiplier and prior designs.

Aloke Saha *et al.* (2018) proposed a novel 7:3 counter for PP accumulation in multiplier. Evaluations with standard 90nm CMOS process revealed superior performance of the proposed multiplier in terms of Power- Delay-Product (PDP) compared to the already reported designs. To the least, the proposed design exhibits 36% and 55% less PDP, compared to the best of the prior design for n-8 and n-16 in the input bitwidth, respectively.

Suganthi Venkatachalam *et al.* (2018) proposed a novel approximate radix-4 Booth Multiplier incorporating probabilistic approach for PP generation and accumulation. Evaluations revealed that the proposed multiplier achieves 41% area reduction and 49% power reduction compared to the conventional Booth multiplier. Analysis with image processing applications revealed superior performance in terms of error metrics compared to prior similar designs.

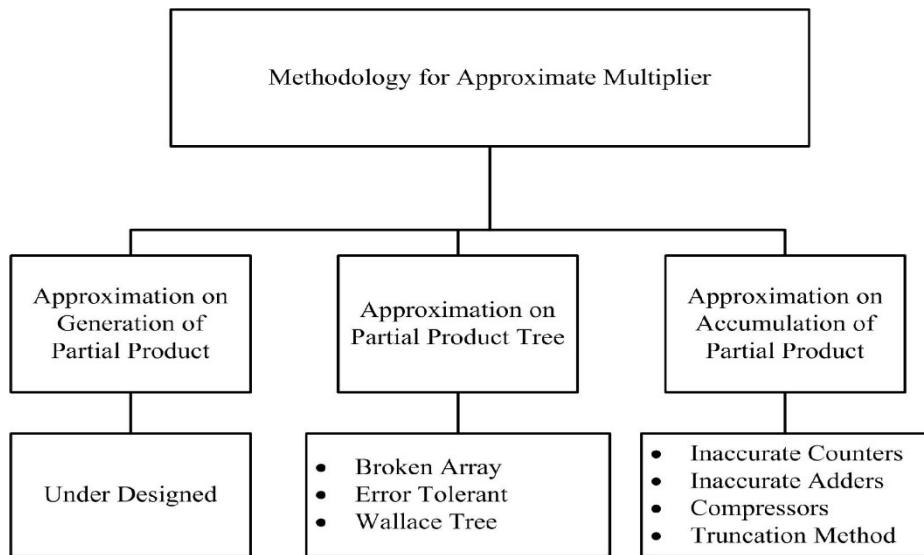
VijeyaKumar et al. (2018) proposed a high speed and energy- efficient error bound fixed width approximate multiplier for unsigned integers. The proposed algorithm reduces error by addition of constant bias at the least significant column of the exact part of multiplier. Moreover the proposed multiplier employs parallel algorithm for PP generation to reduce overall delay. Performance evaluations with structural VHDL model and synthesis with Synopsys design compiler exhibits 66.19% and 36.2% reductions in Chip-Area Ratio (CAR%) and PDP compared to the exact design for an 8X8 multiplication

CHAPTER 3

3.DESIGN OF APPROXIMATE MULTIPLIER

Image processing and multimedia application can tolerate errors and can provide meaningful results. Inexact (approximate) computing techniques have become popular because of its low complexity and less power consumption. Inexact computing produces reasonable result; even it has low accuracy in those applications. In these applications ,multipliers for those applications .Approximation can be applied to every stages of the multiplier. Normally, approximation is applied in anyone stage at a time.

Error performance of the multiplier becomes worst when we apply approximation in more than one stage at the same time. Multi-result. In literature, researchers proposed several approximate multipliers (Kyaw et al. 2010, Bhardwaj et al. 2014, Lau et al. 2009, Venkatesan et al. 2011, Farshchi et al. 2013). Fig.2.11 shows the various methodologies for approximate multiplier. In approximate computing, the value of error rate (ER), Error Distance (ED) and Normal-and Han Jie 2013). Among these, NED is an invariant parameter for any approximate multipliers. Jiang et al. (2016), compared the performance of various approximate multipliers. From the results, use of compressors in the partial product tree gives the lowest error rate, minimum normalized error distance and decent circuit metrics. This section reviews on different approximate multipliers.



In this approach, generation of partial product is approximated to reduce the circuit complexity and delay. This approach was introduced by Kulkarni et al. (2011). 2×2 bit multiplier was designed by altering the logic function of the multiplier. This multiplier has 16 possible combination outputs. Out of 16 combinations, only one output was altered. The Karnaugh map (K-map) of this multiplier is shown in Fig.2.12. Output length of this accurate unsigned multiplier is 4 bits. Instead of using all four bits, three OEth bit when it multiplying “11” and “11”. The actual answer for this input is “1001”. Accurate multiplier uses four output bits to get this answer. Except this output, remaining all other outputs can be represented with the help of three bits.

From the truth table, when “A₁A₀” and “B₁B₀” value are “ll” and “ll”, the output is forced to set “lll”. This is the maximum possible output for 3 bit data ($2^3 - 1 = 7$).

3.1 RELATED APPROXIMATE COMPRESSORS

A compressor is a logic circuit which takes “N” bits as inputs from same or different columns of the multiplier and generates a “Sum” and more than one “Carry” bits as the output.

3.1.1 5:2 COMPRESSOR

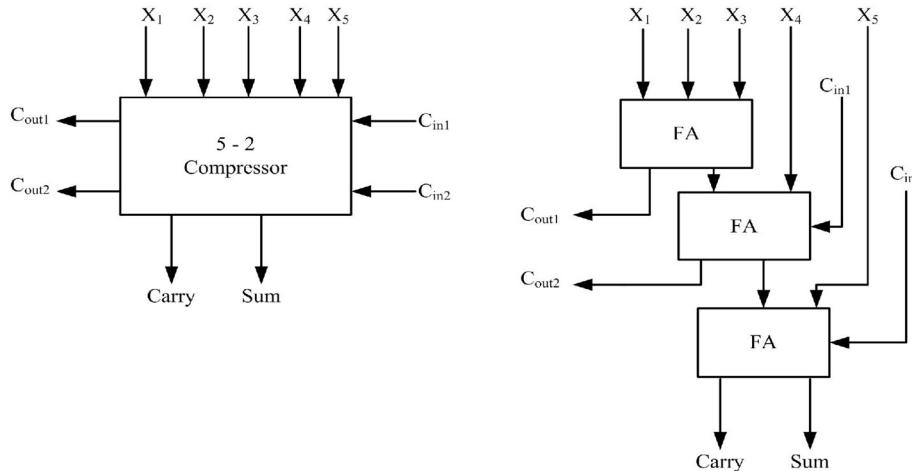


Figure 3.1.1.1 Diagram of 5:2 compressor

Given figure shows the logic diagram of 5-2 compressor. construct the 5-2 compressor. The critical path delay for “Sum” is 4 XOR gates which Carry” is faster than “Sum”. Generation of adjacent carry “Coutl” is independent of previous carry in- put.

3.1.2 6:2 COMPRESSOR

Parandeh-Afshar et al. (2009) proposed this structure. This compressor takes six primary inputs “(E_0, E_1, E_2, E_3, E_4 , and E_5)” and two carry inputs from previous stages “(C_{out0} and C_{out1})” and delivers two primary outputs “(Sum, Carry)” and two adjacent carry outputs “(C_{out0} and C_{out1})”. This structure was implemented with the help of full adders. Fig. shows the implementation of 6-2 compressor using full and half adders. Later, (Ma and Li 2008) implemented the structure using one 4-2 compressor and one gate and 4 XOR gate respectively. Dandapat et al. (2010) proposed and implemented the 6-3 compressor using two full adders and parallel adders. This compressor is not popular because it has the compression ratio of 2 whereas compression ratio of another proposed 6-2 compressor is 3 (Parandeh-Afshar et al. 2009). Higher compression ratio offers minimum partial product reduction stages of the multiplier.

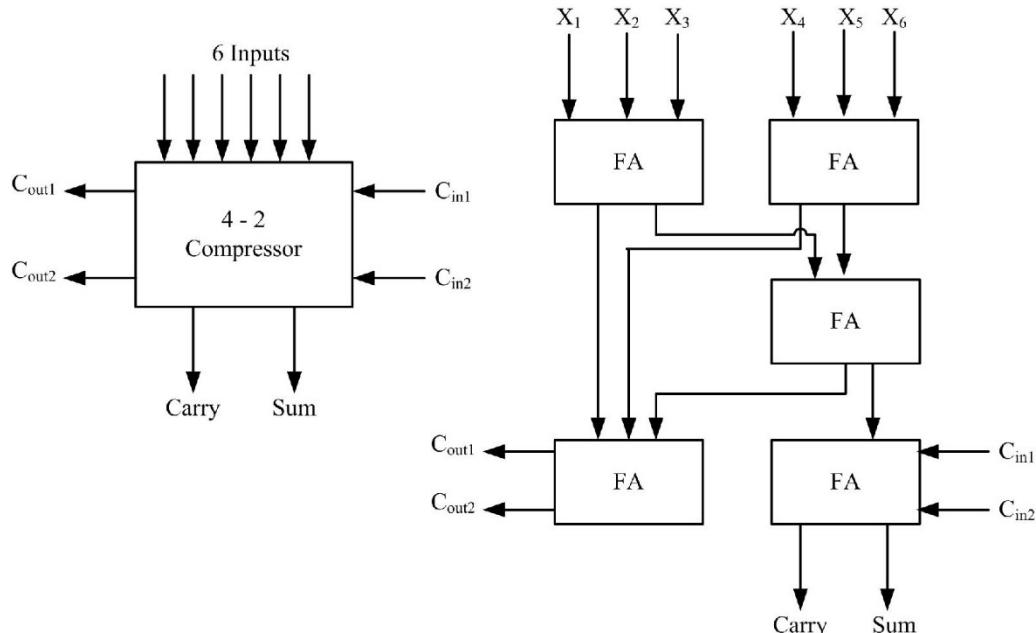


Figure 3.1.2.1 Diagram of 6:2 compressor

3.1.3 7:2 COMPRESSOR

This compressor has seven primary inputs along with two adjacent carry inputs. It delivers two

primary outputs and two adjacent carry to next stages of compressor. Parandeh-Afshar et al. (2009) proposed the direct implementation of this compressor. This implementation has delay of 7 XORs. Rouholamini et al. (2007), Ma and Li (2008) rearranged the expression for 7-2 compressor and worst case delay is reduced to 6 XORs. Also this structure provides less power consumption, PDP and delay. Fig.2.8 shows the structure of 7-2 compressor proposed by (Rouholamini et al. 2007).

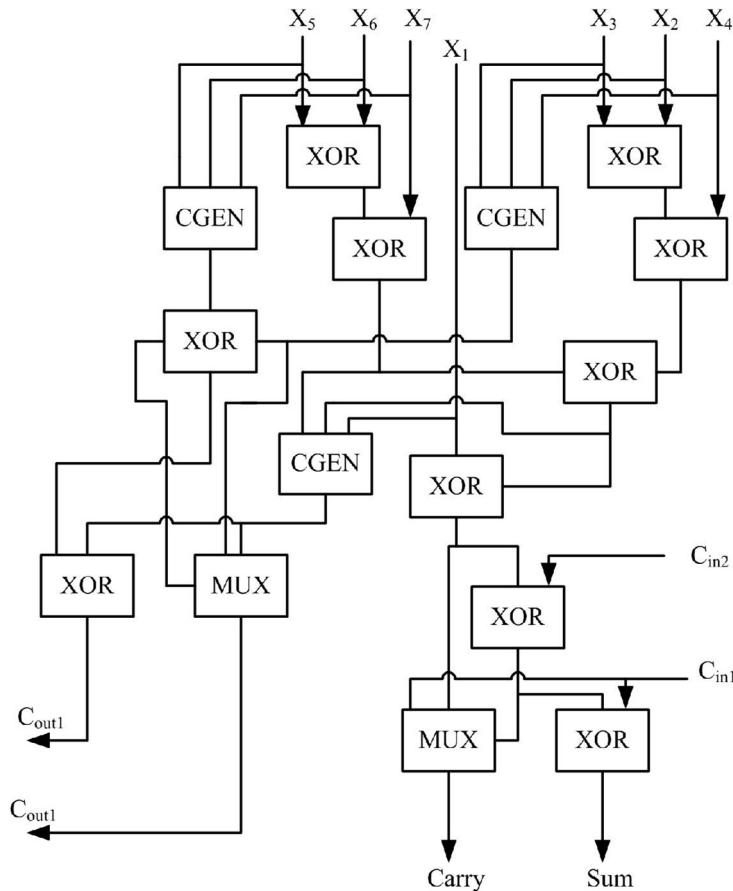


Figure 3.1.3.1 Diagram of 7:2 compressor

‘3.1.4 7:3 COMPRESSOR

Mehta et al. (1991) proposed the 7-3 compressor which implements the logic expressions of 7-3 compressor. Dandapat et al. (2010) implemented the 7-3 compressor using 4-3 compressor and full adder. This implementation reduces 1 XOR gate delay. It has the limitation of irregular layout in

The synthesis. Fig.2 shows the design of this proposed compressor. Veeramachaneni, Avinash, Krishna and Srinivas (2007) Modi-MUXs. Worst case delay of this compressor is 1 XOR

gate and 3 MUXs. Compressors proposed by (Mehta et al. 1991, Dandapat et al. 2010, Veeramachaneni, Avinash, Kr- ishma and Srinivas 2007) has the compression ratio of 2.33. But the compression ratio of proposed compressor is 3.5 (Parandeh-Afshar et al. 2009, Rouholamini et al. 2007, Ma and Li 2008). Highest compression ratio value is preferable in the reduction tree.

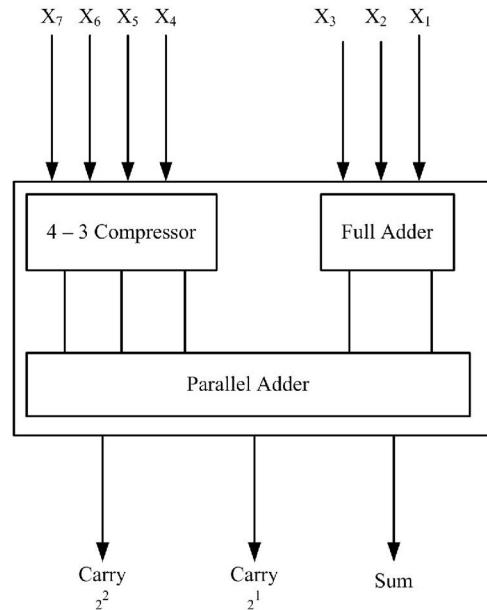


Figure 3.1.4.1 Diagram of 7:3 compressor

CHAPTER 4

INTRODUCTION TO VLSI

4.1 VLSI

VLSI stands for "Very Large Scale Integration". This is the field which involves packing more and more logic devices into smaller and smaller areas.

- Simply we say Integrated circuit is many transistors on one chip.
- Design/manufacturing of extremely small, complex circuitry using modified semiconductor material
- Integrated circuit (IC) may contain millions of transistors, each a few mm in size
- Applications wide ranging: most electronic logic devices

In olden days, when huge computers made of vacuum tubes could occupy an entire dedicated rooms and could do about 360 multiplications of 10 digit numbers in a second. Modern day computers are getting smaller, faster, and cheaper and more power efficient for every progressing second. The electronic miniaturizing started when the occurrence of semiconductor transistor by Bardeen (1947-48) and then the Bipolar Transistor by Shockley (1949) in the Bell Laboratory.

The first IC (Integrated Circuit) was invented by Jack Kilby in 1958, in the form of a Flip Flop our ability to pack more and more transistors onto a single chip has doubled roughly every 18 months, in accordance with the Moore's Law. Such exponential or increasing development had never been seen in any other field and still it is continuing in major areas of research work.

FUTURE OF VLSI:

Generally, VLSI technology is used in the devices like computers, cell phones, digital cameras and any electronic gadget. There are certain key issues that serve as active areas of research and are constantly improving as the field continues to mature. VLSI is dominated by the CMOS technology and much like other logic families, this too has its limitations which have been battled and improved upon since years. By taking the example of a processor, the process technology has rapidly shrunk from 180 nm in 1999 to 60nm in 2008 and now it stands at 45nm and attempts are being made to reduce it for 32nm. As the number of transistors increase, the power dissipation is increasing and also the noise. Heat is generated per unit area. New alternatives like Gallium Arsenide technology are becoming an active area of research; future of VLSI seems to change for every little moment.

History of Scale Integration

- Late 40s Transistor invented at Bell Labs
- Late 50s First IC (JK-FF by Jack Kelby at TI)
- Early 60s Small Scale Integration (SSI)
- 10s of transistors on a chip
- Late 60s Medium Scale Integration (MSI)
- 100s of transistors on a chip
- Early 70s Large Scale Integration (LSI)
- 1000s of transistor on a chip
- Early 80s VLSI 10,000s of transistors on a Chip (later 100,000s & now 1,000,000s)
- Ultra LSI is sometimes used for 1,000,000s
- SSI - Small-Scale Integration (0-102)
- MSI - Medium-Scale Integration (102-103)
- LSI - Large-Scale Integration (103-105)
- VLSI - Very Large-Scale Integration (105-107)

ULSI - Ultra Large-Scale Integration (≥ 107)

ADVANTAGES

These advantages of integrated circuits translate into advantages at the system level are

- **Smaller physical size:** Smallness is often an advantage in itself-consider portable televisions or handheld cellular telephones.
- **Lower power consumption:** Replacing a handful of standard parts with a single chip reduces total power consumption. Reducing power consumption has a ripple effect on the rest of the system: a smaller, cheaper power supply can be used; since less power consumption means less heat, a fan may no longer be necessary; a simpler cabinet with less shielding for electromagnetic shielding may be feasible, too.
- **Reduced cost:** Reducing the number of components, the power supply requirements, cabinet costs, and so on, will inevitably reduce system cost. The ripple effect of integration is such that the cost of a system built from custom ICs can be less, even though the individual ICs cost more than the standard parts they replace.

Understanding why integrated circuit technology has such profound influence on the design of digital systems requires understanding both the technology of IC manufacturing and the economics of ICs and digital systems.

APPLICATIONS OF VLSI

VLSI is having applications in various domains such as electronic systems, medical, communication, digital signal processing. Some of them are listed as below

Electronic system in cars:

Electronic systems now perform a wide variety of tasks in daily life. Electronic systems in some cases have replaced mechanisms that operated mechanically, hydraulically, or by other means; electronics are usually smaller, more flexible, and easier to service. In other cases electronic systems have created totally new applications. Electronic systems perform a variety of tasks, some of them are visible while some are hidden. Personal entertainment systems such as portable MP3 players and DVD players perform sophisticated algorithms with remarkably little energy.

Digital electronics control VCRs:

Digital electronics compress and decompress video, even at high-definition data rates, on-the-fly in consumer electronics. Low-cost terminals for Web browsing still require sophisticated electronics, despite their dedicated function.

Transaction processing system, ATM

Electronic systems in cars operate stereo systems and displays; they also control fuel injection systems, adjust suspensions to varying terrain, and perform the control functions required for anti-lock braking systems.

Personal computers and Workstations

Personal computers and workstations provide word-processing, financial analysis, and games. Computers include both central processing units and special-purpose hardware for disk access, faster screen display, etc.

Medical electronic systems:

Medical electronic systems measure bodily functions and perform complex processing algorithms to warn about unusual conditions. The availability of these complex systems, far from overwhelming consumers, only creates demand for even more complex systems.

4.2 About Verilog HDL

The history of VLSI (Very Large Scale Integration) dates back to the invention of the transistor in 1947, which replaced bulky vacuum tubes and paved the way for semiconductor technology. The development of integrated circuits (ICs) in the late 1950s by Jack Kilby and Robert Noyce led to Small-Scale Integration (SSI) and Medium-Scale Integration (MSI), enabling the integration of hundreds of transistors on a single chip. By the 1970s, Large-Scale Integration (LSI) allowed thousands of transistors, leading to the first microprocessors, such as the Intel 4004 in 1971. The VLSI era emerged in the late 1970s, integrating millions of transistors and revolutionizing computing with CMOS technology. By the 1990s and 2000s, advancements in Ultra-Large-Scale Integration (ULSI) and Deep Submicron (DSM) technology further enhanced chip performance, leading to multi-core processors and high-speed computing. In recent years, breakthroughs in FinFET transistors, 3D ICs, AI-driven VLSI design, and advanced nanometer-scale fabrication (7nm, 5nm, 3nm) have pushed the boundaries of semiconductor technology. Today, VLSI continues to evolve, driving innovations in AI, IoT, quantum computing, and autonomous systems, shaping the future of modern electronics.

4.3 Tool

The main tools required for this project is Xilinx-Vivado 2016 version.

4.3.1 Xilinx

Xilinx software solutions play a pivotal role in the design, simulation, and implementation of digital circuits, enabling engineers to develop highly optimized and efficient FPGA-based systems. As a leader in adaptive computing, Xilinx provides a suite of sophisticated tools that streamline hardware design, facilitate software integration, and enhance system performance across diverse applications.

Among its key offerings, Vivado Design Suite stands out as the primary development environment for modern FPGA and SoC designs. It delivers an advanced synthesis engine, intelligent IP integration, high-level design abstraction, and hardware/software co-design capabilities. Vivado supports high-performance FPGA families, including Zynq, Kintex, and Virtex, and enables seamless development workflows with its built-in HLS (High-Level Synthesis) and AI-driven optimization tools.

For legacy FPGA architectures, Xilinx ISE (Integrated Software Environment) remains a critical tool, offering comprehensive features for HDL-based design, functional verification, and device programming. While now superseded by Vivado, ISE continues to support older FPGA families such as Spartan and Virtex-5, making it an essential tool for maintaining legacy systems.

Additionally, Xilinx has expanded its software ecosystem with Vitis, a unified software development platform tailored for heterogeneous computing. Vitis allows developers to program FPGAs using C, C++, and Python, making FPGA acceleration more accessible for applications in AI, machine learning, and embedded systems.

With the acquisition of Xilinx by AMD, its software ecosystem continues to evolve, integrating FPGA and adaptive computing capabilities with AMD's high-performance CPU and GPU architectures. Xilinx software solutions remain at the forefront of innovation, empowering engineers to develop next-generation computing systems across industries such as telecommunications, automotive, aerospace, and data centers.

4.3.2 Introuction XILINX ISE :

environment. Despite its limitations compared to modern tools, ISE is still valued for its stability, especially in academic and industrial setups that rely on older Xilinx hardware

4.3.3 Xilinx Software

Xilinx software is a collection of professional tools developed by Xilinx (now part of AMD) to support the complete design flow for FPGAs (Field Programmable Gate Arrays), CPLDs (Complex Programmable Logic Devices), and SoCs (System on Chips). These tools are essential for hardware designers and embedded system developers to create, simulate, implement, and deploy digital systems. Xilinx offers three main software suites, each serving different device generations and design needs.

1. Vivado Design Suite

Vivado is the modern and most widely used tool from Xilinx, replacing the older ISE Design Suite for all new FPGA families such as Artix-7, Kintex-7, Virtex-7, Zynq-7000, and UltraScale/UltraScale+ devices. It supports:

- HDL Design (VHDL, Verilog, SystemVerilog)
- Synthesis & Implementation of logic
- IP Integrator for block-based system design
- Vivado Simulator for functional and timing simulation
- Timing analysis, device constraint management, and bitstream generation

Vivado offers a WebPACK edition that is free and supports many entry-level devices, making it accessible for students and beginners.

2. Vitis Unified Software Platform

Vitis is designed for software development on Xilinx SoCs and FPGAs with embedded processors, such as Zynq and Versal platforms. It includes:

- Vitis IDE for C/C++ application development
- Vitis HLS (High-Level Synthesis) for converting C/C++ into RTL
- AI Engine tools for deploying machine learning models on Xilinx devices

Vitis supports hardware-software co-design, allowing a smooth interface between programmable logic and software applications.

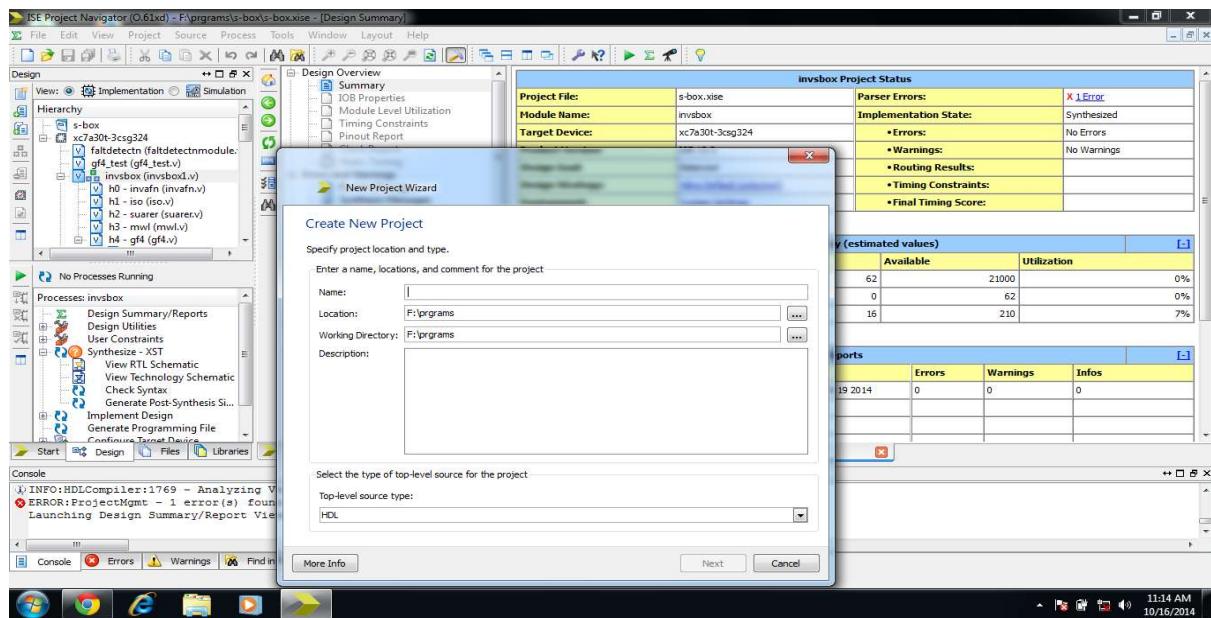
3. ISE Design Suite (Legacy Tool)

ISE is the earlier development environment used for older Xilinx FPGA families like Spartan-3, Spartan-6, and Virtex-5. Though it is no longer updated, it includes tools like:

- Project Navigator (design management)
- XST (Xilinx Synthesis Tool)
- ISim (Simulator)
- **Core Generator** for IP integration It is still used in legacy projects and academic settings where older hardware is involved.

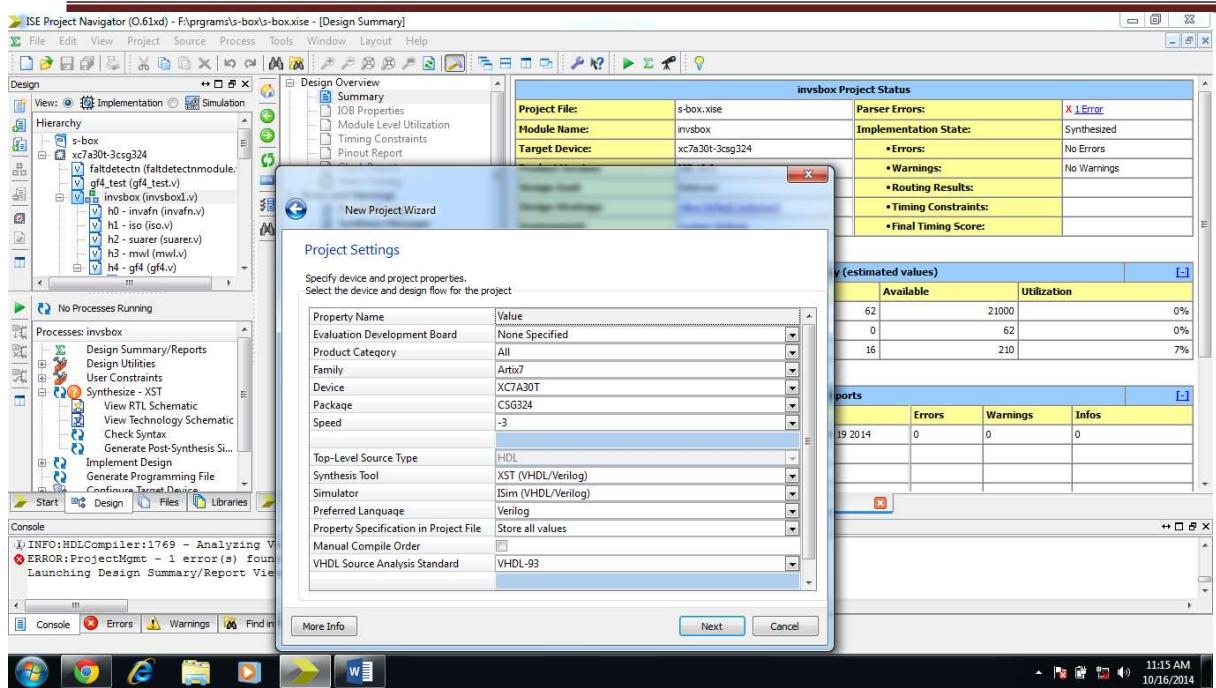
4.3.3.1 Creating a new project

1.To create new project in xilinx we should open the filemenu,click on new project then it will open the dialogbox as below in that type the filename click on next

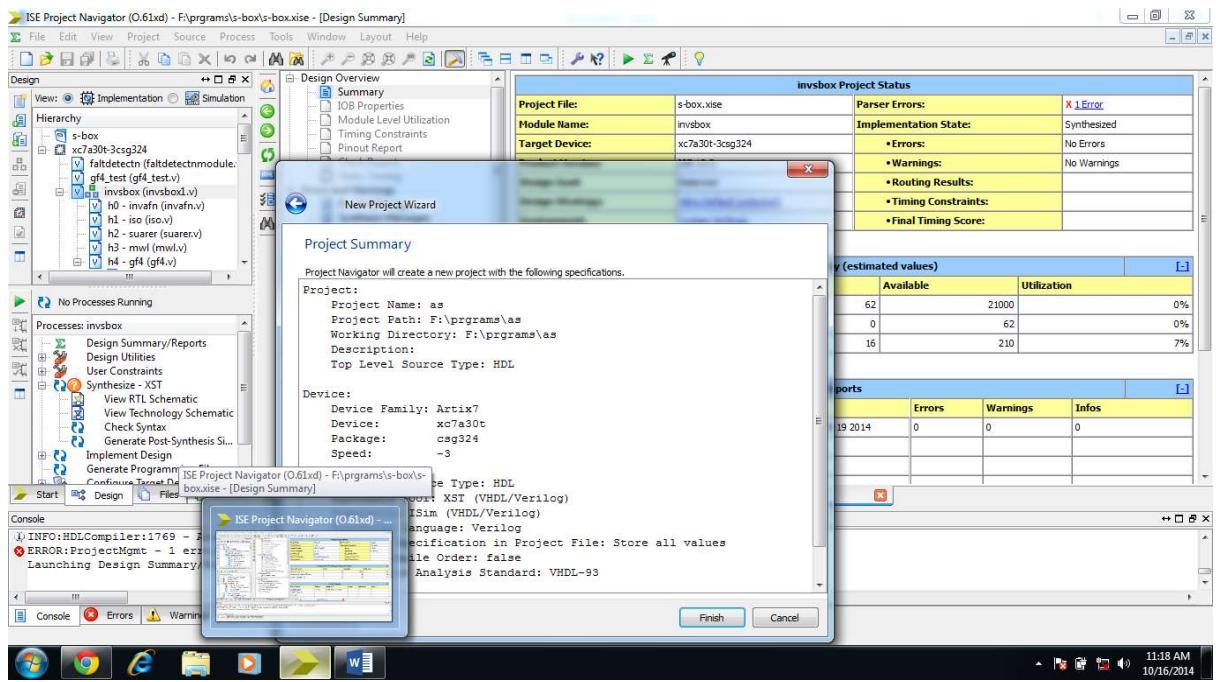


2.Then it displays one more dialogbox which will give us the specifications of the project,click on next

DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE MULTIPLIER USING 15:4 COMPRESSOR

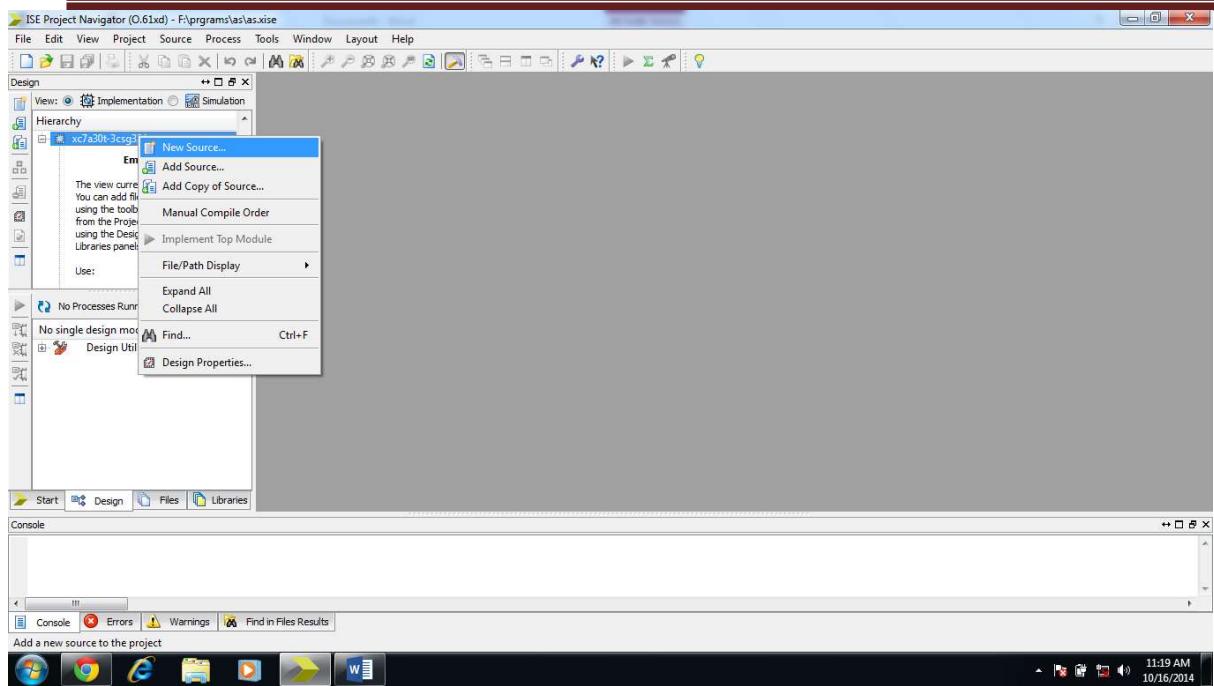


3. Then it again displays a dialogue box as shown below with the created project description and click finish to complete the process of creating new project



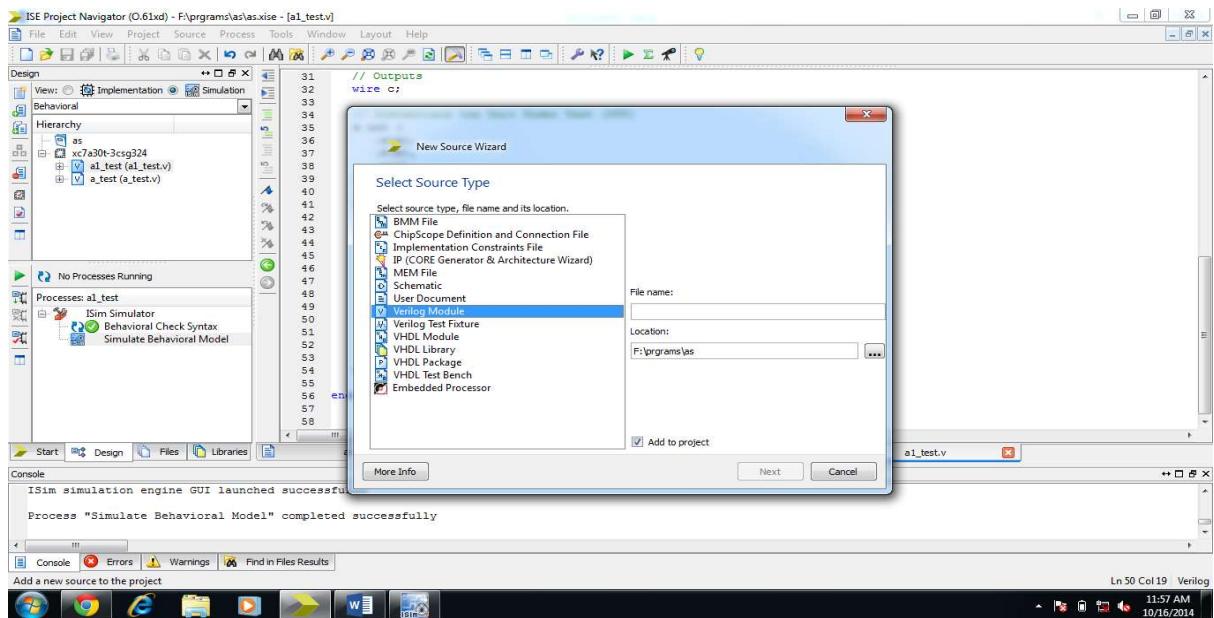
4. Now project with specified name is created then create the verilog files in the project. To create files, right click on the project that will show options like as shown below

DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE MULTIPLIER USING 15:4 COMPRESSOR



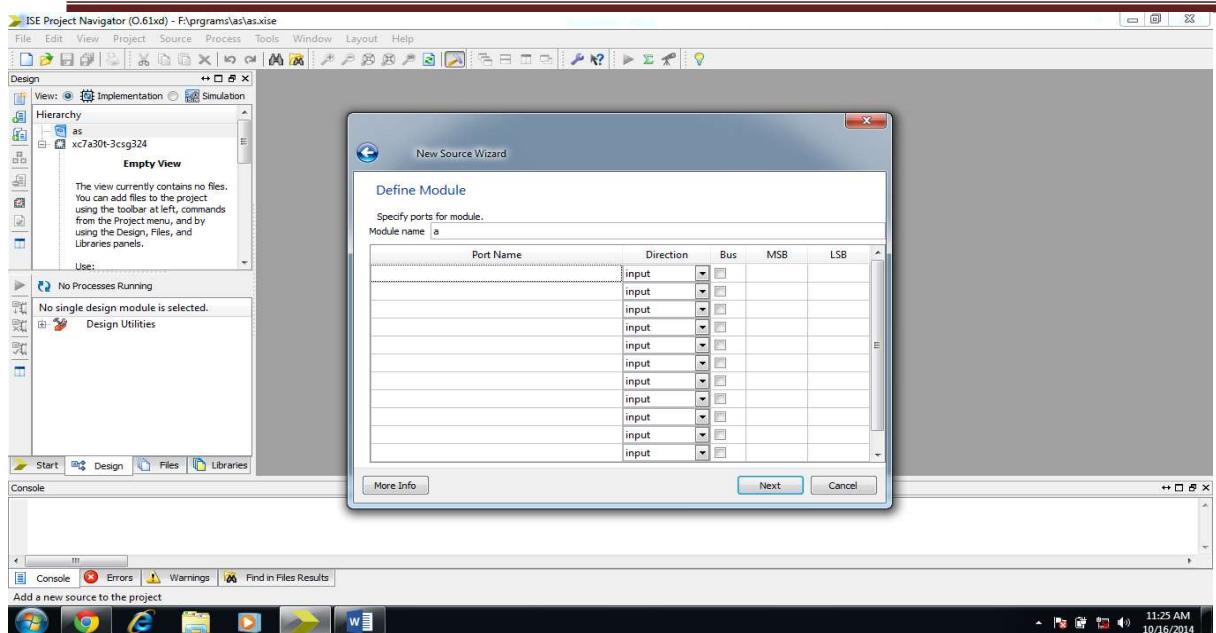
5. From the given options select new source then it displays dialogbox which is containing of list of fileformat now we want to create verlogfile so select veilog module, and give the name to the file.

Then click on next

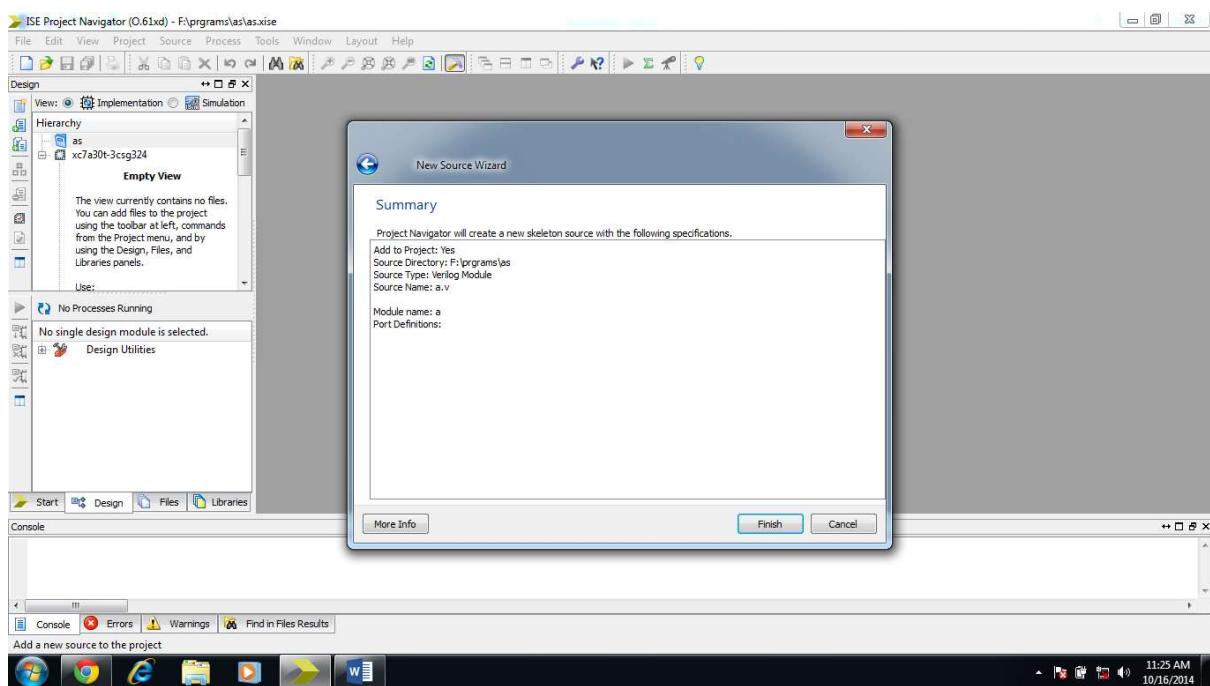


6. Then it will ask us to select inputs,outputs and inout. We can specify our inputs and outputs here else we may also specify as part of programme depend upon the user requirement, click on next

DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE MULTIPLIER USING 15:4 COMPRESSOR

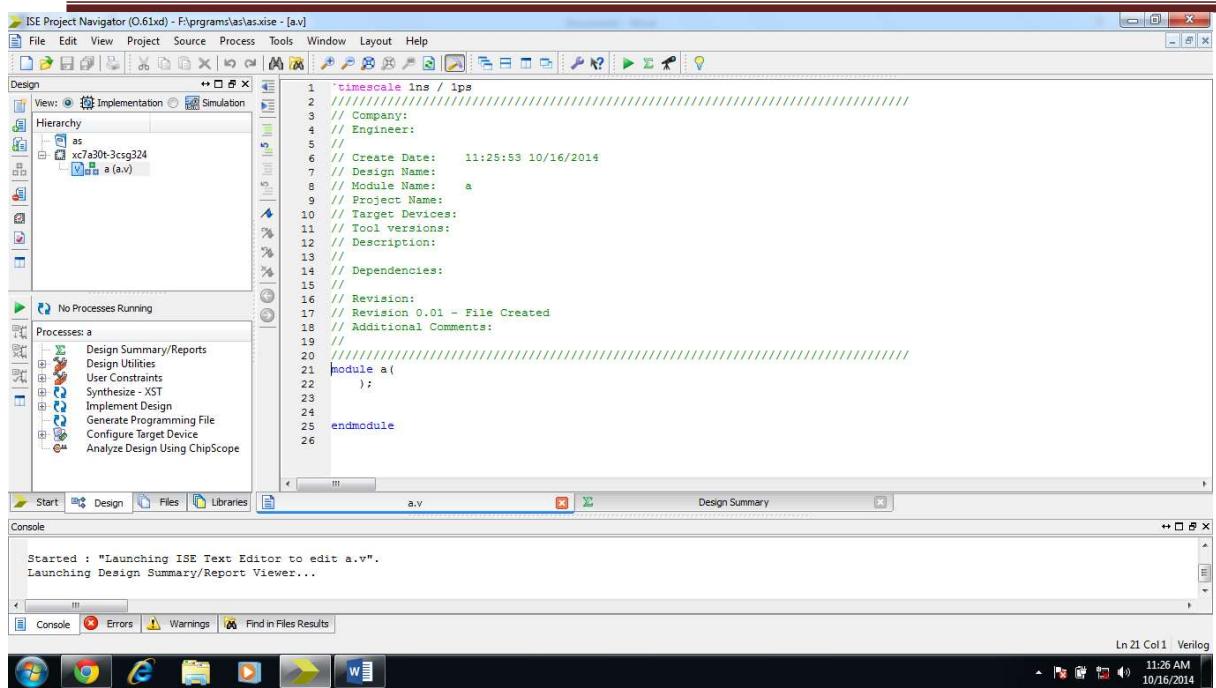


7. It will again displays a dialog box by giving details of filename etc, click on next

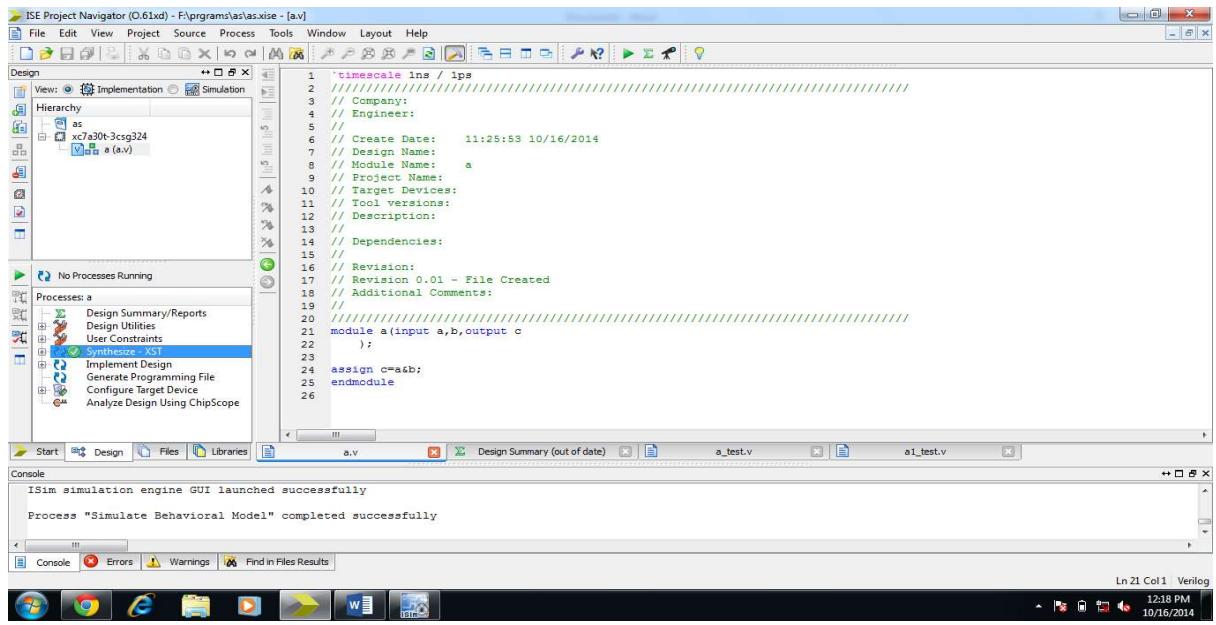


8. It will open a white space in the project window containing filename the double click on the file name so that it will display respective file window ,where we should write the code

DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE MULTIPLIER USING 15:4 COMPRESSOR

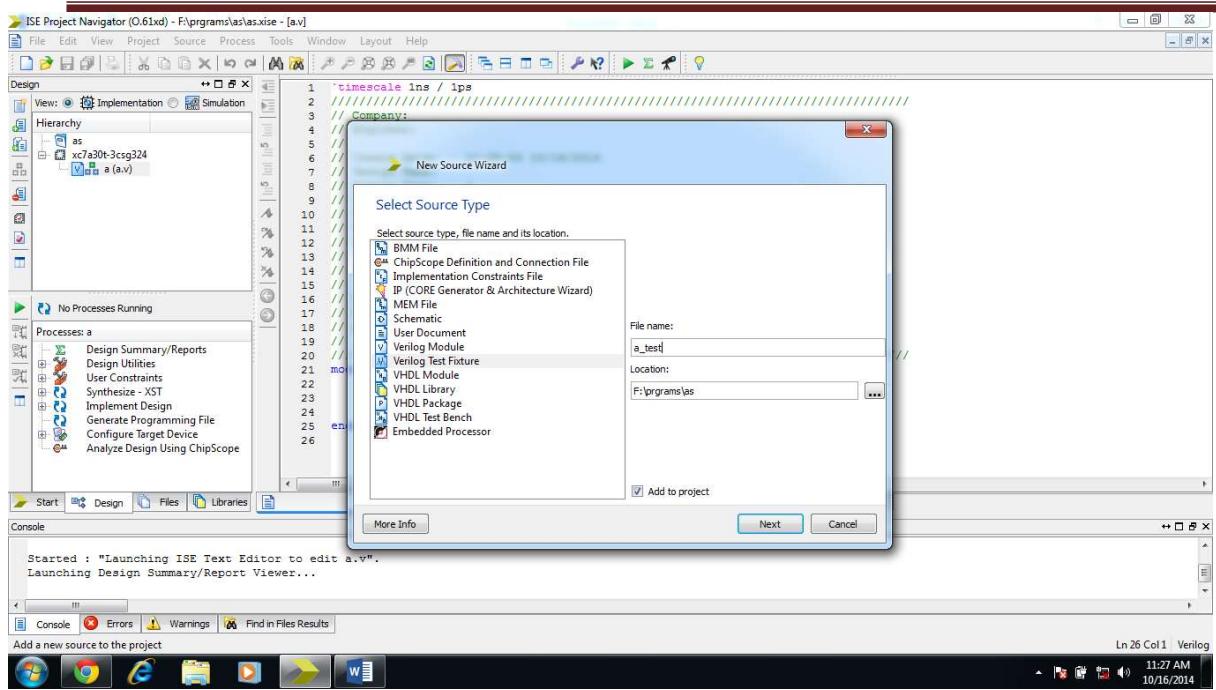


9. After completion writing code select the file name and click on synthesis which will check for errors, if there are any errors in syntax or design errors are checked and shown in the below of file window

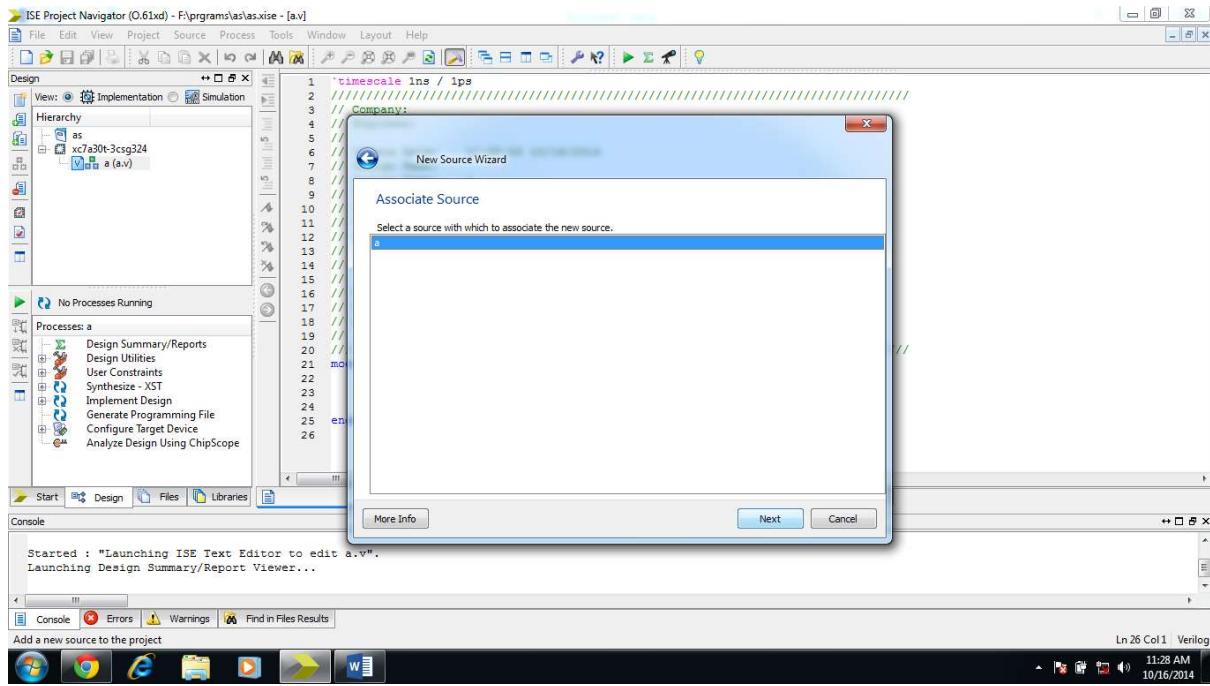


10. After sucessful synthesis we should have to create tesh bench file with extension as test,for that again riht click on the file name as shown below,give filename

DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE MULTIPLIER USING 15:4 COMPRESSOR

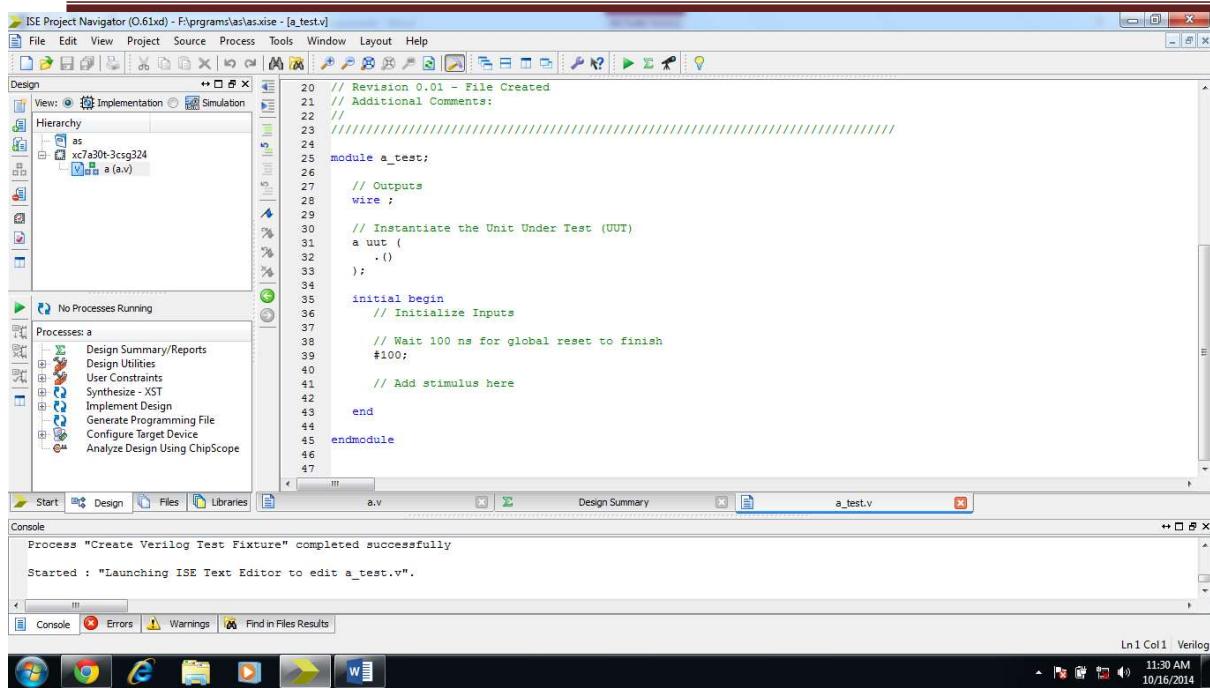


11.If there are list files then select file for which we are creating the test bench. Click on next

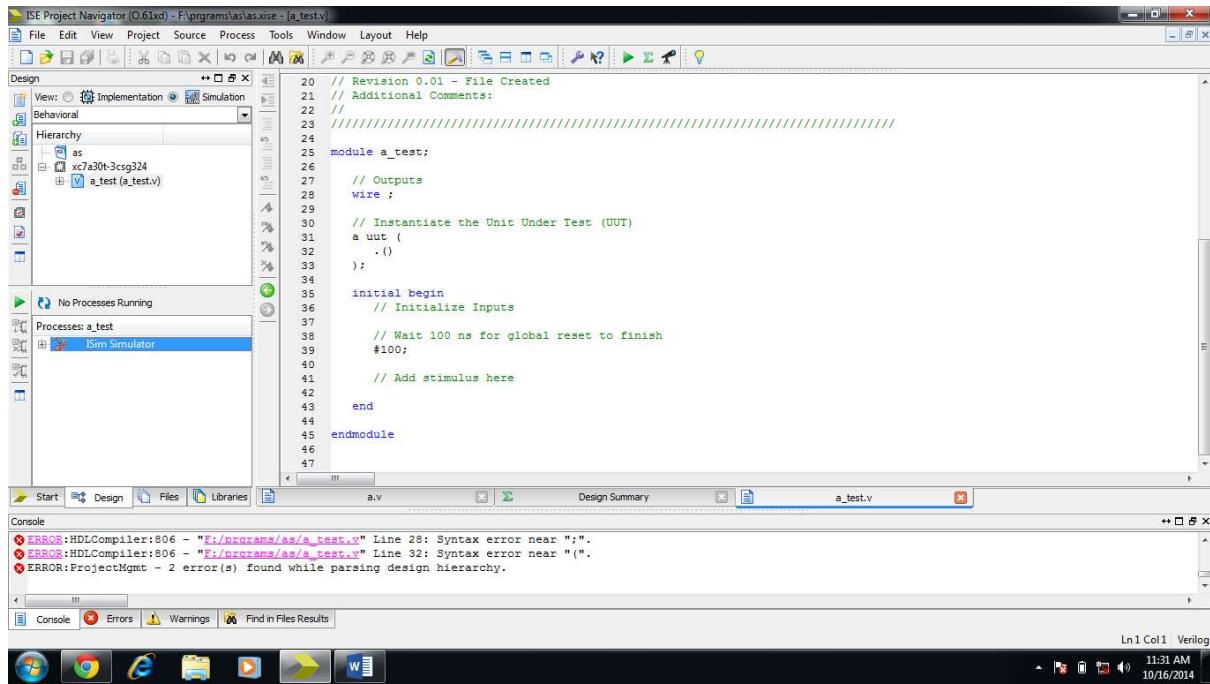


12.It again gives a testbench file in the project window, then give required inputs

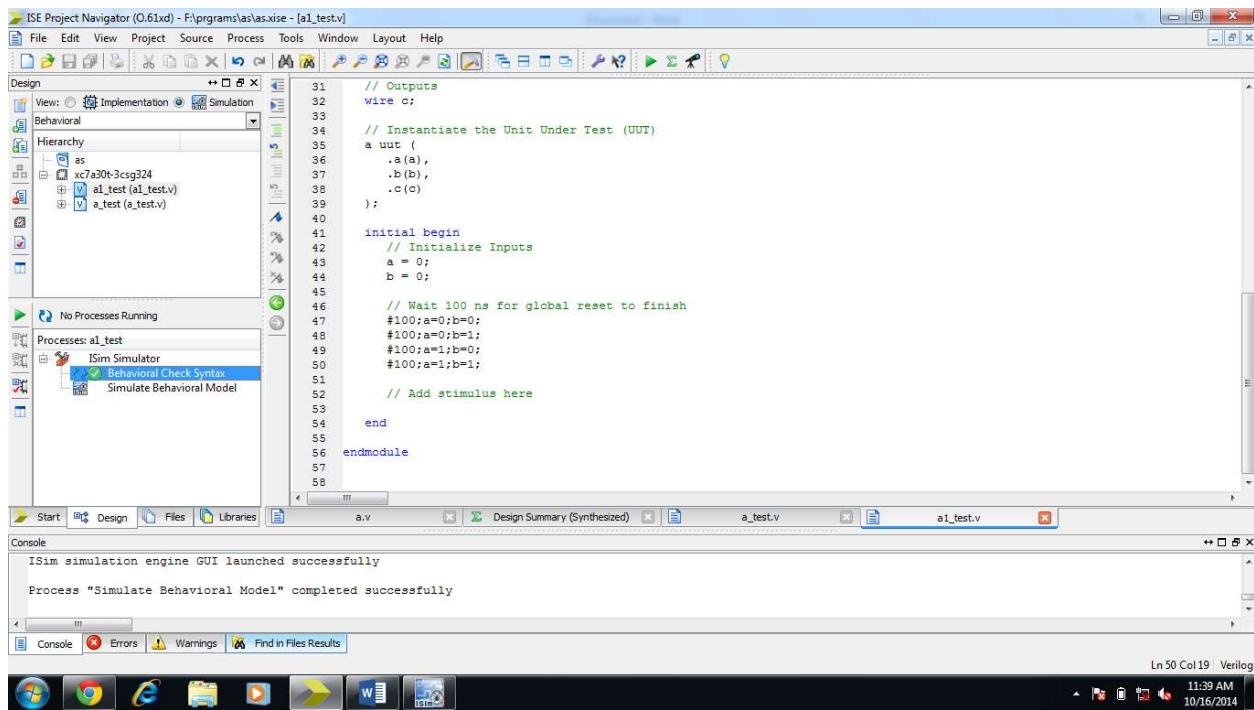
DESIGN AND ANALYSIS OF LOW POWER APPROXIMATE MULTIPLIER USING 15:4 COMPRESSOR



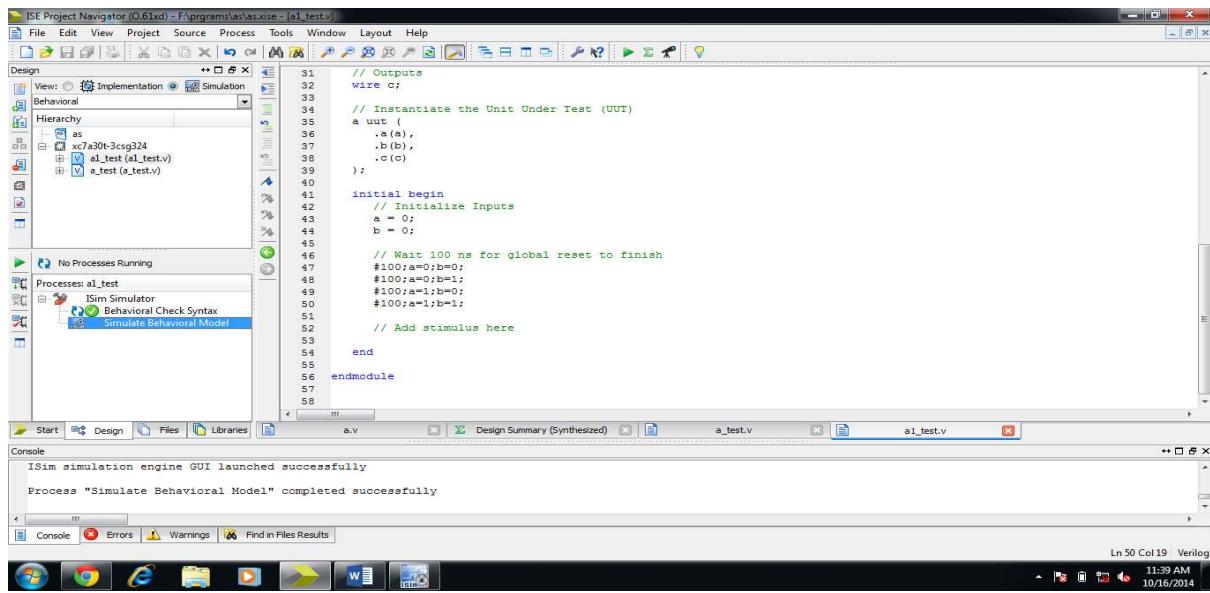
13.select simulation from the view bar in the project window above the hierarchy window as follows.



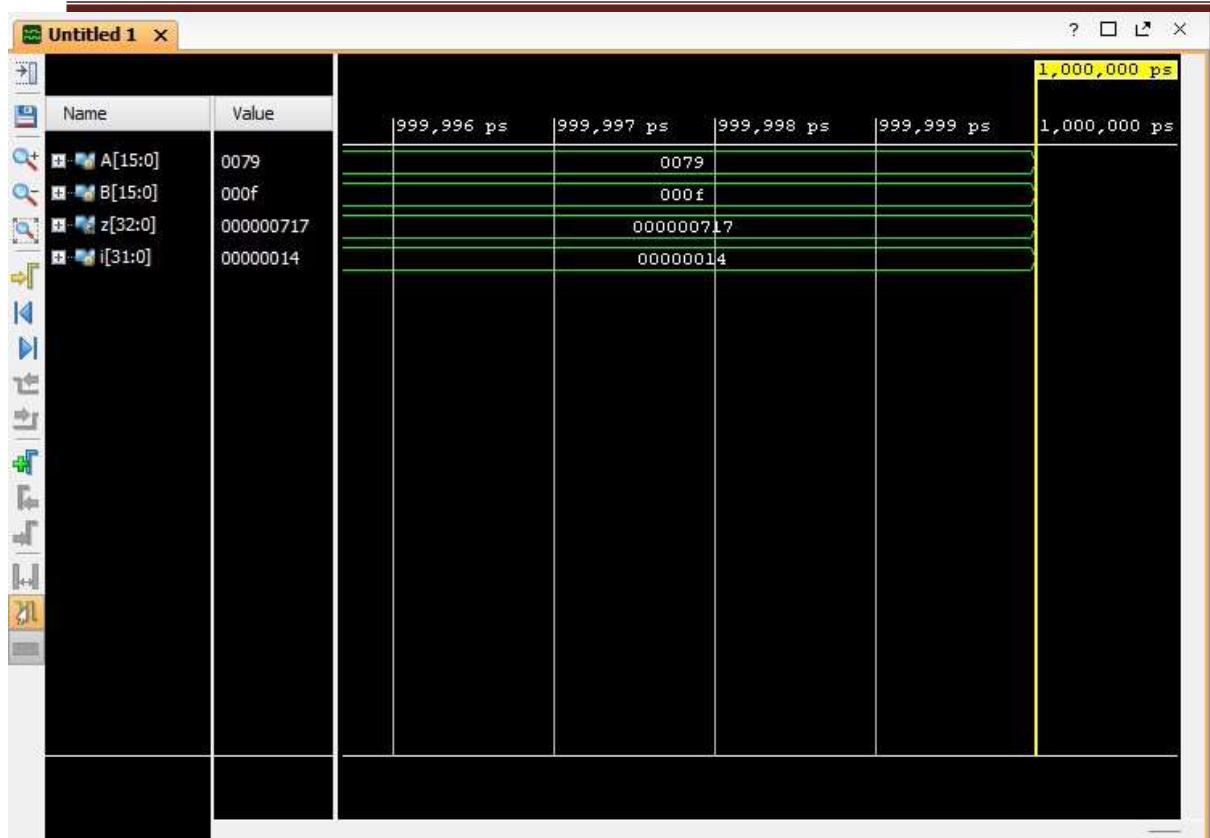
14. Double click on Isim Simulator it will expand as follows click on behavioral check syntax and it will check for syntax errors in test bench file



15.click on simulate behavioral model, it will displays wave form for in response to the inputs given in the test bench file



16.That wave form window having option to zoom out, zoom in to analyze the wave form clearly in order to understand behavior of design



CHAPTER 5

EXISTING METHOD

5.1 Block diagram of Existing method

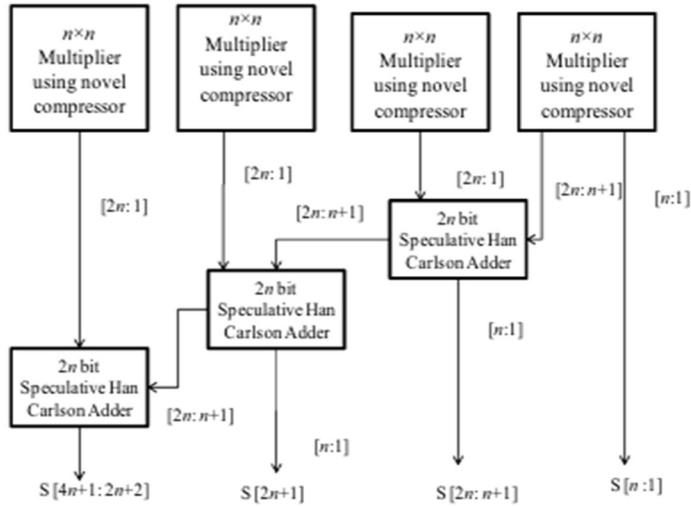


Figure 5.1.1 Block diagram of Han-carlson adder

There are three primary steps in parallel multipliers. The formation of partial products is the first stage, which is carried out by AND gates. The second stage is the reduction of the partial products which can be done by using approximate novel compressors. The final addition of the partial products is done using approximate Han-Carlson parallel prefix adder , in the third stage. The second stage, out of the three, is the most power-hungry stage. As a result, enhancing the second stage improves the multiplier's performance while lowering its cost. As the bit-width of approximated Han-Carlson parallel prefix adder increases the power consumption and area utilization is also double when compared with the previous bit-width. In approximate Han-Carlson parallel prefix adder the power and area are increasing exponentially. The implementation of the proposed approximated multiplier using novel compressors and parallel-prefix adder increases the performance of the circuit. In this paper, the authors implemented architectures of the approximate multiplier by using the approximate Han-Carlson adder and multiplier using novel compressor.

5.2 Existing compressors

5.2.1 3:2 Compressor

A 3:2 compressor is a fundamental building block used in high-speed arithmetic circuits, especially in multipliers and adders. It is designed to reduce three input bits into two output bits, optimizing the carry propagation and improving overall performance in digital circuits.

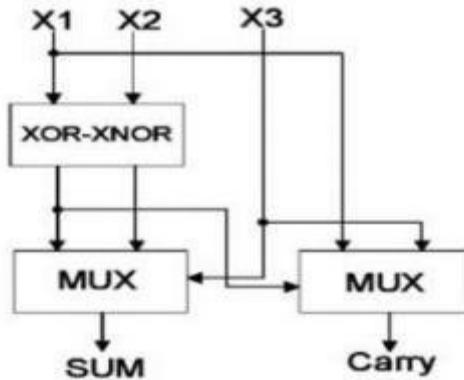


Figure 5.2.1.1 Block diagram of 3:2 compressor

Architecture of 3:2 Compressor

A 3:2 compressor takes three input bits (X, Y, Z) and compresses them into two output bits, a Sum (S) and a Carry (C), similar to a full adder but optimized for parallel computations.

Truth Table of 3:2 Compressor

Logic Equations

The 3:2 compressor outputs can be defined as:

- Sum (S) = $X \oplus Y \oplus Z$
- Carry (C) = Majority(X, Y, Z) = $(X \cdot Y) + (Y \cdot Z) + (X \cdot Z)$

This structure is similar to a full adder, but in compressor-based designs, multiple compressors are cascaded for efficient parallel computation in multipliers.

Operation of 3:2 Compressor

The 3:2 compressor is widely used in multipliers and multi-operand adders. Instead of processing addition serially, it reduces three bits to two, allowing multiple compressors to operate in parallel,

significantly improving speed.

- In a Wallace Tree or Dadda Tree multiplier, multiple 3:2 compressors are used to minimize partial product stages, reducing delay.
- Compressors help in designing low-power and high-speed arithmetic circuits in VLSI applications.

Advantages of 3:2 Compressor

- Reduces Carry Propagation Delay – Allows faster addition in multipliers and accumulators.
- Improves Speed – Reduces the number of logic stages compared to traditional adders.
- Efficient for Large Bit-width Arithmetic – Commonly used in high-performance multipliers (e.g., Wallace and Dadda Trees).
- Scalable Design – Can be extended to higher-order compressors (e.g., 4:2, 5:2, etc.) for further efficiency.

Disadvantages of 3:2 Compressor

- More Hardware than a Basic Adder – Requires extra logic gates compared to simple adders.
- Implementation Complexity – Needs optimized VLSI layout to minimize power and area consumption.
- Not Ideal for Small Circuits – More beneficial in high-speed applications rather than basic arithmetic operations.

5.2.2 4:2 compressor

A 4:2 compressor is a specialized digital circuit commonly used in high-speed arithmetic operations like multiplication and addition. It plays a crucial role in hardware implementations of multipliers, especially in Wallace tree and Dadda tree architectures, which aim to reduce the time delay in summing partial products. The name "4:2 compressor" indicates that it takes 4 primary inputs (A, B, C, D) and 1 carry-in (Cin) from the previous stage, and produces 2 outputs — a sum and a carry, along with a carry-out ($Cout$) to the next significant bit position.

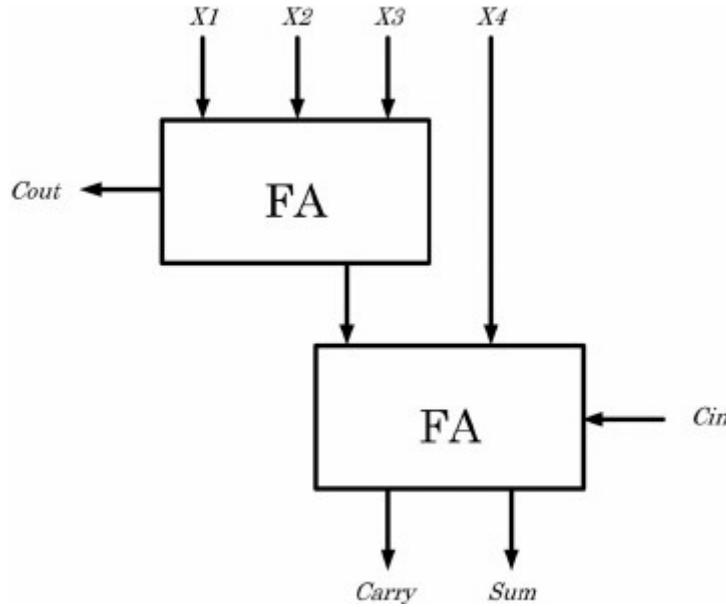


Figure 5.2.2.1 Block diagram of 4:2 compressor

Operation of 4:2 Compressor:

The main function of a 4:2 compressor is to compress five input bits (A, B, C, D, Cin) into three outputs (Sum, Carry, Cout). This operation is typically done in two stages:

1. First, three inputs (say A, B , and C) are fed into a full adder to produce an intermediate sum and carry.
2. Second, this intermediate sum is added to the remaining input (D) and the carry-in (Cin) using another full adder.
3. The final outputs are:
 - o Sum: the result of the second full adder,
 - o Carry: the carry from the second adder (fed to the same bit position in the next stage),
 - o Cout: the carry from the first full adder (propagated to the next higher bit position).

This parallel processing reduces the number of logic levels and increases the speed of computation.

Advantages of 4:2 Compressor:

- **High Speed:** It reduces the critical path delay by minimizing the number of sequential

additions.

- Efficient Arithmetic Operations: Especially useful in fast multipliers and arithmetic circuits.
- Parallelism: Allows multiple operations to occur at once, improving overall performance.
- Hardware Optimization: Reduces the number of components needed in a multiplier tree, making it more compact.

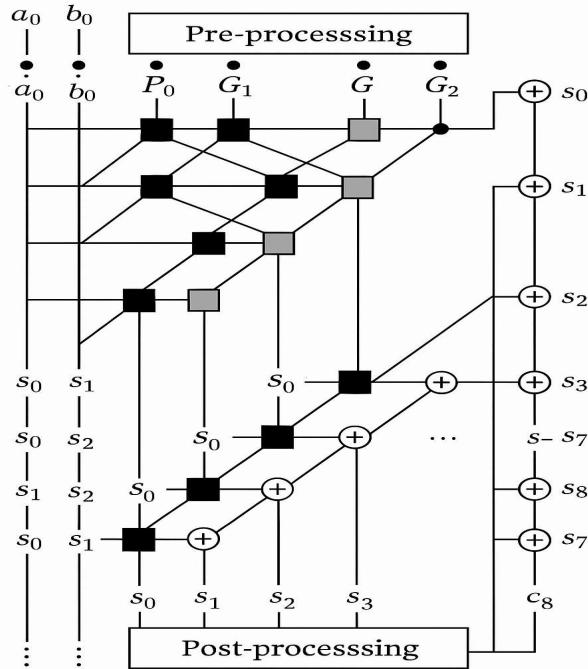
Disadvantages of 4:2 Compressor:

- Complex Design: More complicated than basic adders like half and full adders.
- Increased Power Consumption: Due to more gates and switching activity.
- Difficult Debugging: Analyzing and tracing errors in the circuit is harder compared to simple arithmetic units.
- VLSI Challenges: In large-scale integration, routing and placement of compressor circuits can be challenging.

5.3 Existing adder

5.3.1 Han-carlson adder

The Han-Carlson Adder (HCA) is a parallel-prefix adder that efficiently balances speed, power consumption, and area complexity. It is a hybrid design that takes inspiration from the Kogge-Stone Adder and Brent-Kung Adder, aiming to reduce wiring congestion while maintaining a high-speed addition process.

**Figure 5.3.1.1 Han-carlson adder**

Architecture of Han-Carlson Adder

The Han-Carlson Adder consists of three main stages:

1. Generate and Propagate Calculation (G/P Stage)

- Each bit of the two input numbers generates G (Generate) and P (Propagate) signals based on:

$$G_i = A_i \cdot B_i$$

$$P_i = A_i \oplus B_i$$

- These signals help determine whether a carry will be generated or propagated at each bit position.

2. Parallel-Prefix Computation (Carry Propagation Stage)

- The carry computation process is structured in a way that blends the fast parallelism of Kogge-Stone with the efficient serial structure of Brent-Kung.
- It starts with a fully parallel prefix network in early stages (similar to Kogge-Stone) and transitions to a more serial structure in later stages (like Brent-Kung).
- The prefix tree structure reduces the total number of computational nodes, leading to a lower gate count and better power efficiency.

3. Final Sum Computation (Sum Stage)

- The sum is computed using the carry signals and the propagate signals through:

$$S_i = P_i \oplus C_{i-1}$$

- This step finalizes the addition process, providing the correct sum output.

Operation of Han-Carlson Adder

The Han-Carlson Adder works in the following steps:

1. Input Stage

- The input binary numbers are processed to generate the G (Generate) and P (Propagate) signals.

2. Carry Computation Stage

- The carry network processes carry signals using a hybrid prefix tree.
- It uses parallel prefix computation in early stages to quickly generate intermediate carries.
- In later stages, a more serial and structured approach is used to reduce wiring complexity.

3. Sum Computation Stage

- Once all carry bits are computed, the final sum is obtained using an XOR operation with the propagate signals.

Advantages of Han-Carlson Adder

Speed Optimization

- Provides faster addition than Brent-Kung Adder and is close in speed to Kogge-Stone.
- The hybrid prefix structure ensures efficient carry propagation.

Lower Wiring Complexity

- Reduces the number of interconnects compared to Kogge-Stone, lowering wiring congestion.

Power and Area Efficiency

- Uses fewer logic gates than Kogge-Stone, making it more power-efficient.
- The hybrid prefix approach leads to a good balance between speed and area usage.

Scalability

- Suitable for large-bit-width additions, making it ideal for VLSI designs and ALUs

- Works well in high-performance computing environments.

Disadvantages of Han-Carlson Adder

More Complex than Brent-Kung

- While optimized, it still requires more gates than the Brent-Kung Adder, making it less area-efficient.

Higher Power Consumption than RCA or CLA

- Even though it is more efficient than Kogge-Stone, it still consumes more power than simpler adders like Ripple Carry Adder (RCA) or Carry Look-Ahead Adder (CLA).

Implementation Complexity

- The hybrid prefix tree structure requires careful design, making it slightly harder to implement in comparison to simpler adders.

5.3.2 Types of Adders

In digital electronics, adders are fundamental combinational logic circuits used to perform binary addition. They play a crucial role in arithmetic and logic units (ALUs), microprocessors, digital signal processors (DSPs), and various embedded systems. As technology has advanced, several types of adders have been developed to address different requirements such as speed, power consumption, area, and complexity.

Ripple Carry Adder (RCA)

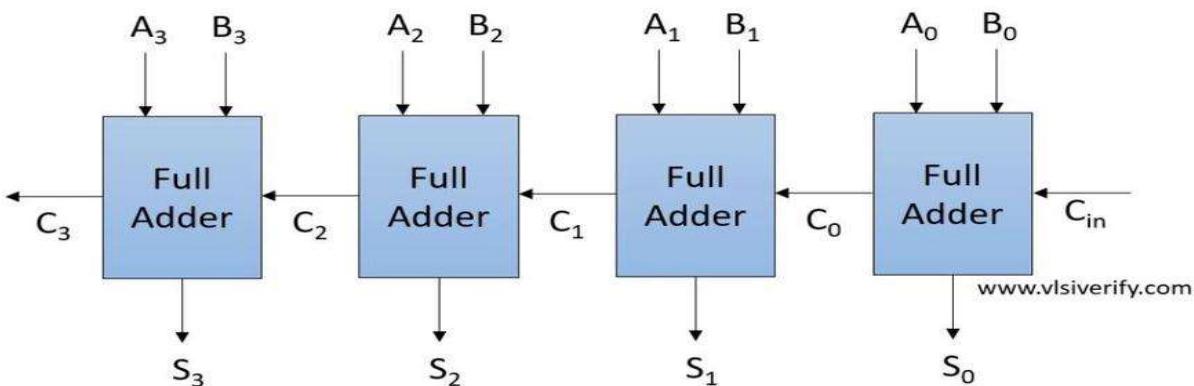


Figure 5.3.2.1 4 bit Ripple carry adder

The basic unit of ripple carry adder is full adder. It can be constructed by connecting full adders in cascaded, with the carry out of the previous 1-bit full adder is given as carry-in to the next 1-bit full adder in the chain. In this cascaded structure, carry out propagates or ripples through the circuit. Ripple carry adder occupies smaller area on the chip and offers high performance to random input data. The delay of the ripple carry adder depends on the length of the propagation path. Due to this reason, RCA is not suitable for circuits with non-random input operands. In the ripple carry adder, the output is known only after the carry of the previous stage is produced.

Thus, the sum of the most significant bit is only available after the carry signal has rippled through the adder from the least significant stage to the most significant stage which is worst case addition. As the result, the final sum and carry bits will be valid after a considerable delay. The delay associated with RCA is given as:

$$t_{RCA} = n t_{FA}$$

Where n is the number of 1-bit FAs connected in RCA and t_{FA} is the delay associated with 1-bit full adder circuit. This critical delay increases linearly with number of bits (n).

Carry Look-Ahead Adder (CLA)

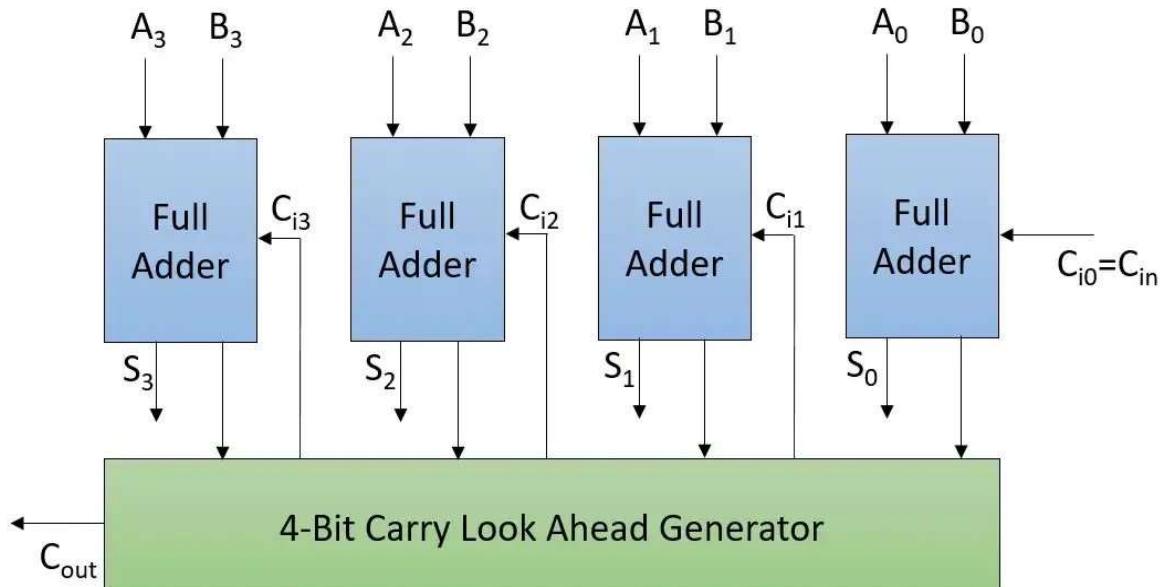


Figure 5.3.2.2 4 bit carry look ahead adder

The major problem associated with ripple carry adder is its delay which increases with number of bits or depends upon the propagation path from least significant bit to most significant bit. In ripple carry, it is required that carry should be passed through all lower bits to compute the sum for higher bits.

Therefore, for fast applications, a better design is required which can be achieved by carry look-ahead adder (CLA). CLA solves the problem of delay in RCA by calculating the carry signal in advance based on the input signal. Therefore, CLA provides lower delay than RCA at the price of more complex hardware and large area on the chip. Generate and propagate logic is used in the CLA.

The main advantage of CLA is that carry delay and sum delay are independent of the number of bits one need to add. The disadvantage of the carry look-ahead adder is that the carry logic becomes complicated for more than 4-bits.

Carry Select Adder (CSL)

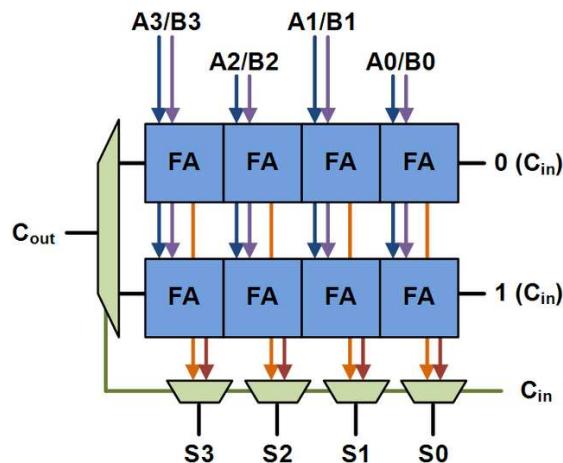


Figure 5.3.2.3 carry select adder (CSL)

In a ripple carry adder, every full adder cell has to wait for the carry in signal before generating carry out, which is time consuming. One way to get rid of this problem is to assume both possible values of the Cin, i.e., $C_{in} = 0$ and $C_{in} = 1$.

After that, it is required to evaluate the result for both possibilities in advance. After knowing the correct value of Cin, the correct result will be chosen with the help of 2:1 MUX. This idea is implemented in the design of carry select adder.

Therefore, carry select adder computes two results in parallel. Each result is computed for two different values of carry in. The carry select adder is simple and fast. Addition of two n-bit (a,b) numbers is performed by breaking the input into two blocks.

For each carry save block, sum and carry values are propagated for both carry-in =0 and carry-in = 1. The actual carry-out value is then fed into a multiplexer that picks the correct sum and carry-out for the next block. When the two phases of carry-in are equal, the total gate delay is minimal.

Save Adder (CSA)

In carry save adder, carries are saved as partial carries rather than propagated. These partial carries are added to the next operand during the next addition. One can accelerate each addition by postponing the carry propagation. The carry save adder adds up numbers by a series (multiple operand addition), followed by a carry propagate (carry propagate) addition. A carry save adder sums up a partial sum and partial carry from the previous stage as well as operand and produces a new partial sum and partial carry.

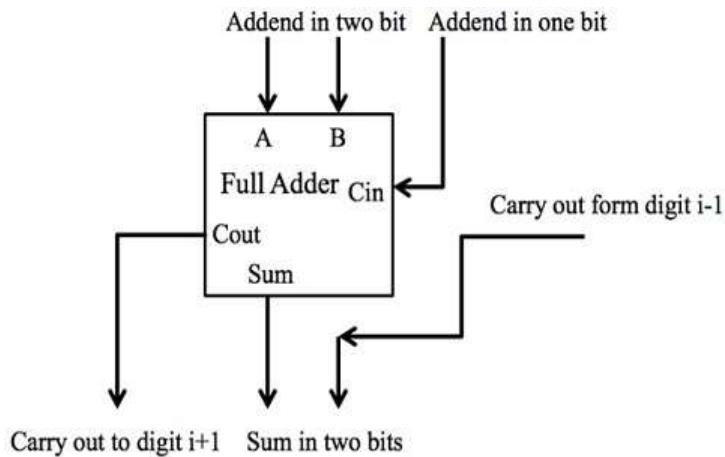


Figure 5.3.2.4 Carry Save adder (CSA)

The Han-Carlson Adder is a type of parallel adder that is designed to improve the performance of binary addition by reducing the carry propagation delay. It builds upon the principles of the carry-lookahead adder (CLA), but with a more efficient structure.

CHAPTER 6

PROPOSED METHOD

6.1 Block diagram of proposed multiplier

Multiplier has become the most significant portion of the digital circuits which carries the majority of operation at system level. It is a circuit utilized in digital electronics, kind of a computer, to multiply two binary numbers. Binary adders are used construct a multiplier. An effective multiplier ought to have the subsequent features:

- **Accuracy:** An efficient multiplier have got to produce proper results.
- **Speed:** Multiplier must achieve operations at tremendous speed.
- **Area:** A multiplier must cover a lesser amount of slices and LUTs.
- **Power:** Multiplier ought to use minimum power.

There are number of several multipliers and some of them are:

1. Booth multiplier,
2. Array based multiplier
3. Wallace tree structure multiplier,
4. Combinational circuit multiplier,
5. Sequential circuit multiplier.

A WTM is an effective hardware which makes use of electronic circuit that products numbers. In this design, a Wallace tree multiplier since it has the advantages of superior processing speed and minimal power utilization.

There are three stages for a multiplication process usually occurs are:

1. Generation of Intermediate partial products
2. Reduction of them
3. Addition at the end.

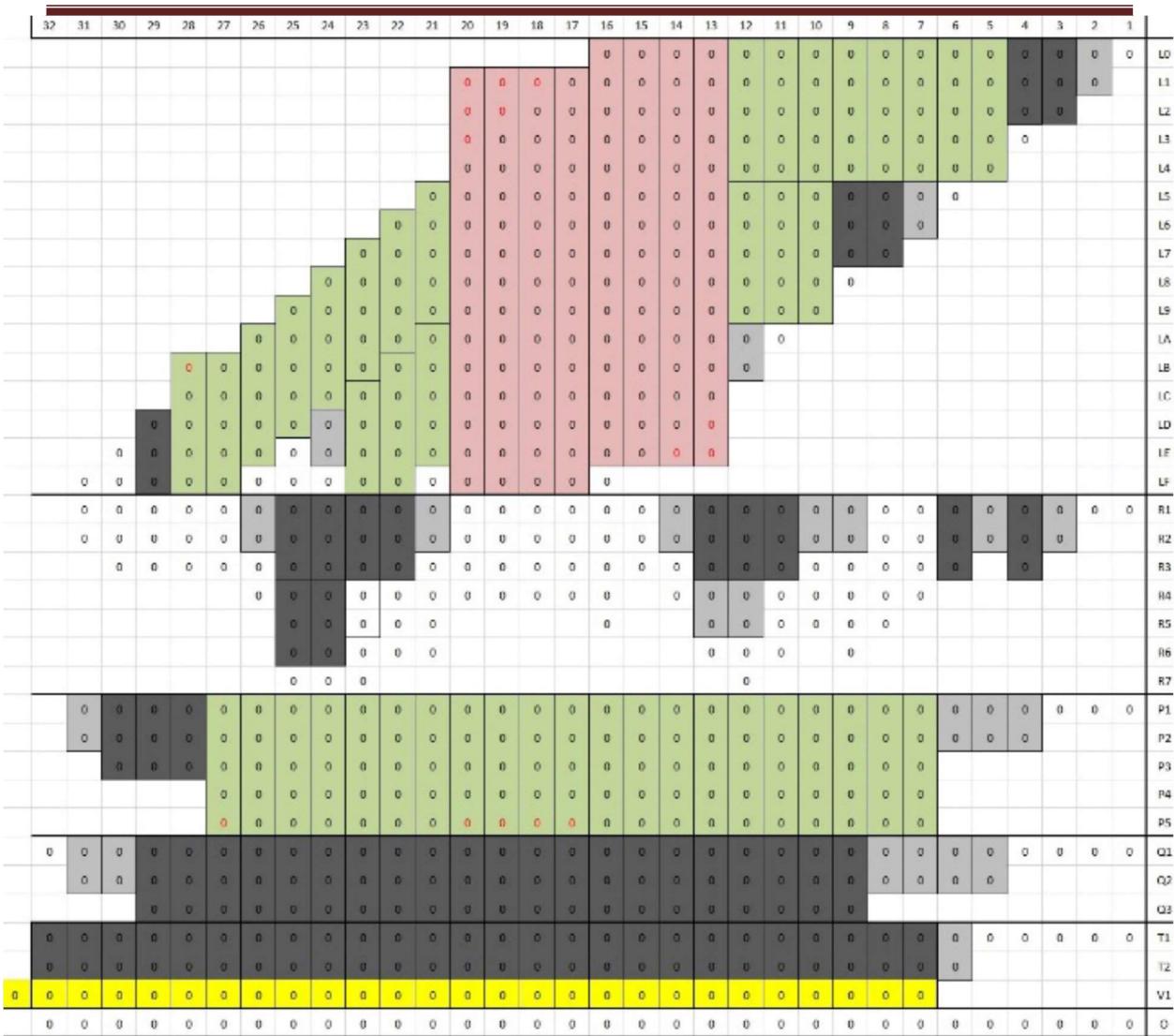


Figure 6.1.1 Structure of 16-bit Wallace tree multiplier using 15:4 compressor

Fig describes the Configuration or Structure of 16-bit WTM using 15-4 compressor and modified approximate full adder. In this model, every dot denotes a partial product. Between the column number 13 to column number 20, a 15-4 compressor is employed and there 13 partial products and two zero are included in the 13th column in order to have 15 inputs for the compressor to perform. By the same way, a zero is added. Column number 13th to 20th needs to be performed using approximate compressor. Various half adder are involved in the second stage of the multiplier, modified approximate full adders and 5:3 compressors. For every individual bit, it is brought down to successive stages of column without any chance of additional actions. Till only two rows stay, the method of reduction is persisted.

6.2 Proposed Compressor

6.2.1 5:3 Compressor

15-4 compressor or compactor is made up of two 5-3 compressors in order to obtain 3 respective compressed outputs. The 5-3 compressor applies 5 initial inputs such as S01, S02, S03, S04, and S05 produces three outputs namely S_01, S_02 and S_03. In the same way, another 5-3 compressor is used to obtain the results from carry inputs namely C_1, C_2, C_3, C_4 and C_5. Yield at compressor is being determined by the number of 1's in the place of the input and further implemented by the property of counter.

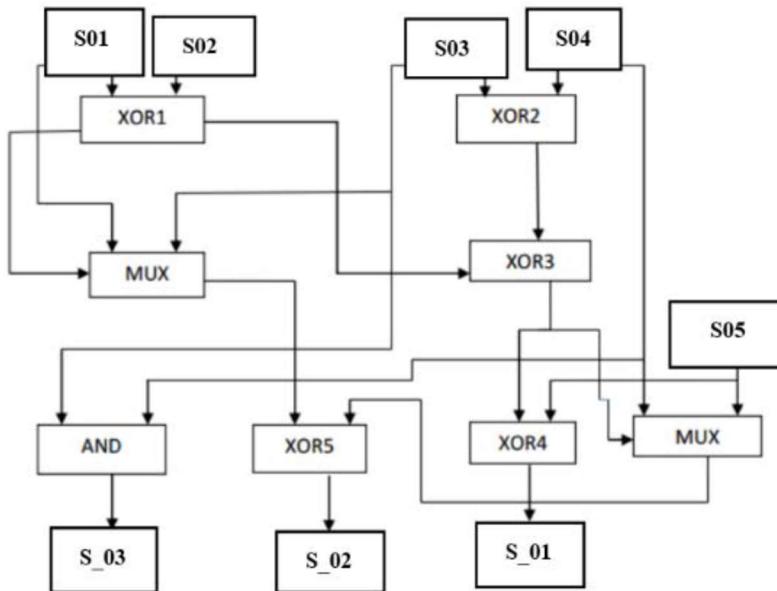


Figure 6.2.1.1 Block representation of 5-3 compressor

Advantages of 5:3 Compressor

1. Reduces Addition Stages – Compresses multiple bits in a single step, reducing the number of required adders.
2. Faster Computation – Decreases propagation delay compared to conventional adders.

3. Efficient Multiplier Design – Used in high-speed multipliers to optimize the partial product summation.
4. Lower Power Consumption – Reduces the number of transitions in arithmetic circuits, leading to power savings.
5. Better Hardware Utilization – Saves area by replacing multiple full adders with a single compressor.

Disadvantages of 5:3 Compressor

1. Increased Complexity – More complex logic gates compared to standard full adders.
2. Higher Wiring Overhead – More interconnections can lead to routing congestion in ASIC and FPGA implementations.
3. Difficult Timing Optimization – Delays vary depending on the logic levels, requiring careful design.
4. Not Always Optimal for Small Circuits – Overhead may not justify its use in low-power, small-scale applications.
5. Potential Glitches – In asynchronous designs, timing variations can cause glitches affecting performance.

6.2.2 15:4 compressor

Compactors or compressor are merely used as an adding circuit. This compressor has fifteen inputs ($C_0 - C_{14}$) and it delivers four outputs ($B_0 - B_3$). The compressor has 5 modified approximate full adders at the initial phase, 2 5-3 compressors in the secondary phase and the last phase has a KSA. Sum and Carry is generated out of the given inputs. One of the two 5-3 compressors obtains the sum bits of all the modified approximate full adders. Likewise, the other compressor obtains the carry bits of all the modified approximate full adders. A compressor adder delivers lowered delay on standard adders applying all the half adders with modified approximate full adders. Yields of intermediate compressors provided as input for the KSA. With the use of KSA at last stage the output is obtained. Compressors are used in the reduction of quantity of gates also the amount delay when compared to the other adders.

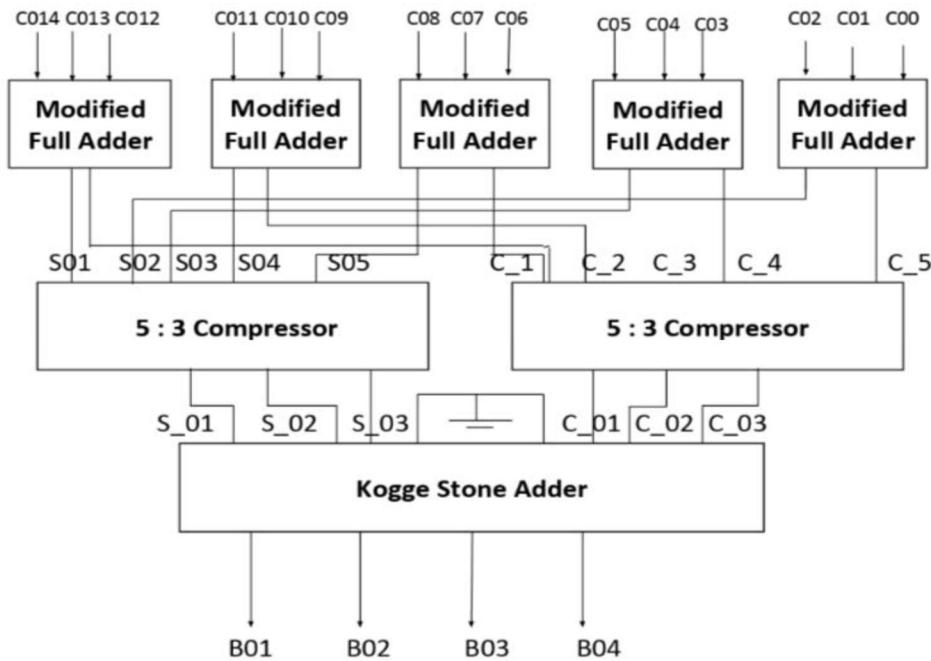


Figure 6.2.2.1 Block Representation of Approximate 15-4 Compressor using Modified Approximate Full Adder

The Approximate compressor of 15-4 involves three segments. 1st segment consists of negative modified approximate 3 input full adders, whereas 2nd segment consists of two 5-3 compressors and 3rd segment the 4-bit KSA.

Advantages of 15:4 Compressor

1. **Faster Computation** – Reduces the number of addition stages, leading to lower delay.
2. **Efficient Area Usage** – Uses fewer logic levels than ripple adders, making it more compact.
3. **Reduced Power Consumption** – Since fewer additions are needed, dynamic power consumption decreases.
4. **Better Pipelining** – Facilitates high-speed arithmetic designs by reducing propagation delays.
5. **Improved Performance in Multipliers** – Useful in Wallace Tree and Dadda multipliers for faster accumulation of partial products.

Disadvantages of 15:4 Compressor

1. **Complex Design** – Requires more logic gates compared to simpler adders.
2. **Higher Wiring Overhead** – Increased interconnections can lead to routing congestion.
3. **More Power for Small Designs** – Not always optimal for low-power applications due to added control logic.
4. **Difficult Implementation** – May require advanced circuit techniques for efficient layout in ASIC or FPGA.
5. **Propagation Delay Variability** – The delay is not always uniform, affecting timing analysis in high-frequency circuits.

Modified Approximate Full Adder:

Modified approximate full adders have been used in the initial stages of the entire architecture. There are exactly 5 number of modified approximate full adders in the design. This is used in the area of conventional full adders to reduce the gate counts using approximation concept. The modified approximate full adder uses only one OR gate, one EX-OR gate and one AND gate in total to obtain the desired function of a full adder thereby reducing the number of gate counts and other parameters like power, area. These modified approximate full adders after its functioning produces 5 equivalent sum outputs and 5 equivalent carry outputs.

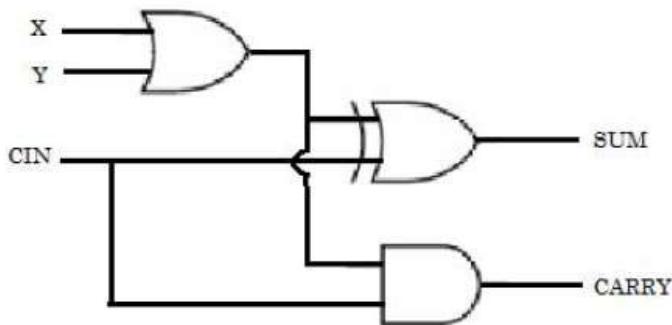


Figure 6.2.2.2 Logic Structure of the Modified 3-bit Approximate Full adder

6.3 Proposed Adder

6.3.1 Kogge Stone Adder

The KSA is one of the well-known parallel prefix adder. KSA is found to be the fastest in terms of addition in design perspective. The KSA acquires extra area when compared to Brent–Kung adder, but KSA has lesser fan-outs at each phase, which rises execution. In this design, a 4-bit KSA is used. These type of Adders are mostly categorized in 3 types.

- A. Pre- processing stage
- B. Carry Generation section
- C. Final processing or Post processing.

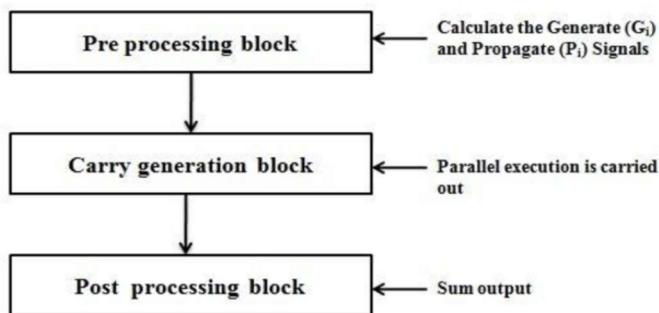


Figure 6.3.1.1 Functions of basic KSA

Stages of Kogge-Stone Adder Architecture

The architecture consists of three main stages:

1.Pre-Processing Stage (Generate & Propagate Calculation)

Each bit i of two input binary numbers AAA and BBB generates two signals:

- Generate: $G_i = A_i \cdot B_i$ (Indicates if a carry is generated at bit i)
- Propagate: $P_i = A_i \oplus B_i$ (Indicates if a carry is propagated)

2.Prefix Processing Stage (Carry Computation)

- The carries are computed in a hierarchical manner using a tree structure.
- Uses black and gray cells to compute the carry signals:
 - Black Cells: Compute both generate and propagate values.
 - Gray Cells: Compute only generate values.
- The carry signals are computed using:

$$G_{i:j} = G_i + (P_i \cdot G_j)$$

$$P_{i:j} = P_i \cdot P_j$$

- The tree-based computation reduces the carry propagation delay to $O(\log N)$.

3.Post-Processing Stage (Sum Calculation)

The sum bits are computed using:

$$S_i = P_i \oplus C_{i-1}$$

Where $C_{i-1}C_{i-2}\dots C_1C_0$ is the carry computed in the prefix stage.

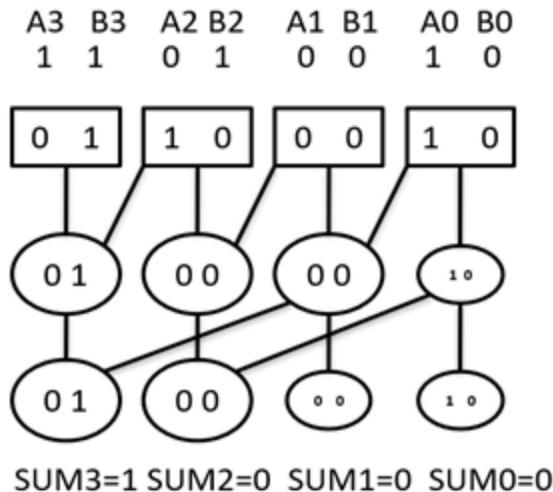


Figure 6.3.1.2 Block representation of KSA

Advantages of Kogge-Stone Adder

1. Fastest Adder – It has the shortest delay due to parallel prefix computation.
2. Regular Structure – Makes it ideal for VLSI implementations.
3. Scalability – Works well for larger bit-widths (e.g., 32-bit, 64-bit adders).
4. Logarithmic Delay – The carry propagation delay is $O(\log N)$, making it much faster than ripple carry adders ($O(N)$).
5. Parallel Computation – Uses parallel prefix computation, reducing latency significantly.

Disadvantages of Kogge-Stone Adder

1. High Area Overhead – Requires a large number of logic gates, leading to high hardware complexity.
2. More Power Consumption – More gates mean higher power dissipation.

3. Routing Congestion – The large number of connections leads to routing issues, especially in FPGA and ASIC designs.
4. Not Always Optimal – For smaller bit-widths (like 4-bit, 8-bit), simpler adders like Brent-Kung or Ripple Carry Adders may be more efficient

CHAPTER 7

RESULT AND DISCUSSION

7.1 Code for 16-bit proposed multiplier

```
module fifteen_4_approx(x,sum);

input [14:0]x;
output [3:0]sum;
wire [4:0] s,c;
wire [2:0]p,q;
wire c1,c2,c3;
wire [3:0]a,b;

fa_approx fa1(x[0],x[1],x[2],s[0],c[0]);
fa_approx fa2(x[3],x[4],x[5],s[1],c[1]);
fa_approx fa3(x[6],x[7],x[8],s[2],c[2]);
fa_approx fa4(x[9],x[10],x[11],s[3],c[3]);
fa_approx fa5(x[12],x[13],x[14],s[4],c[4]);

five_three_comp_approx c1000(p[2],p[1],p[0],s[4],s[3],s[2],s[1],s[0]);
five_three_comp_approx c20(q[2],q[1],q[0],c[4],c[3],c[2],c[1],c[0]);
assign a[0]=p[0];
assign a[1]=p[1];
assign a[2]=p[2];
assign a[3]=1'b0;
assign b[0]=1'b0;
assign b[1]=q[0];
assign b[2]=q[1];
assign b[3]=q[2];
kogge K1(a[3:0],b[3:0],1'b0,sum);
endmodule
```

7.2 Implementation

16-bit results

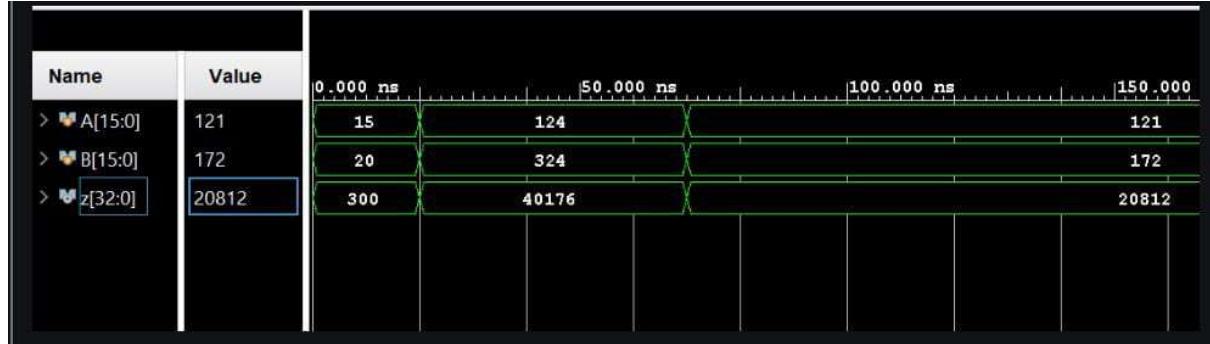


Figure 7.2.1 simulation Result 16-bit multiplier

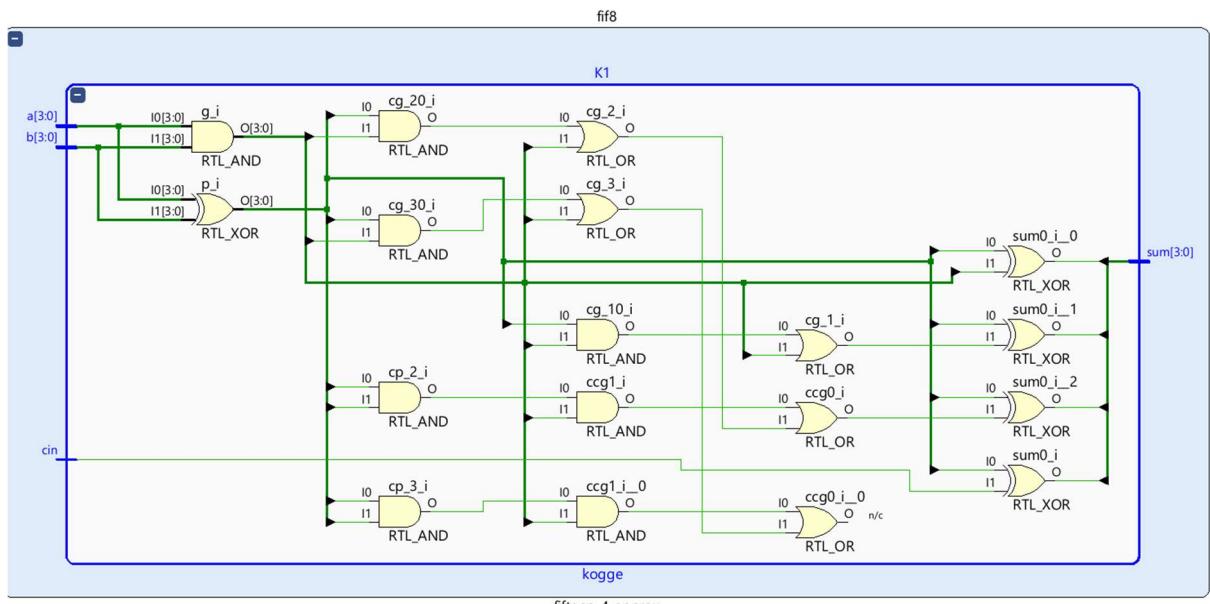


Figure 7.2.2 Schematic Diagram of 15:4 compressor



Figure 7.2.3 Delay report of 16-bit multiplier

The total delay of the paths ranges from 21.665 ns to 22.739 ns, with net delay contributing the majority of this delay—approximately 15.6 to 16.7 ns. Logic delay values are relatively moderate, ranging from around 5.8 to 6.4 ns. The high net delays suggest that routing or interconnects are causing significant delay, which could be due to long wire lengths, high fan-out, or suboptimal placement. These paths may not be critical to the timed portions of the design, but if left unconstrained, they can become a source of unexpected timing issues.

The critical parameters observed in the report include Total Delay, Logic Delay, and Net Delay. The total delay across these paths ranges approximately from 21.66 ns to 22.73 ns, with the net delay—representing routing and interconnect delays—constituting the majority of the path delay. The logic delay, which accounts for the delay due to combinational elements like gates, ranges from 5.88 ns to 6.42 ns, whereas net delay lies between 15.64 ns and 16.69 ns. The dominance of net delay indicates that these paths are heavily influenced by routing congestion, long wire lengths, or high fan-out, which may lead to suboptimal timing performance.

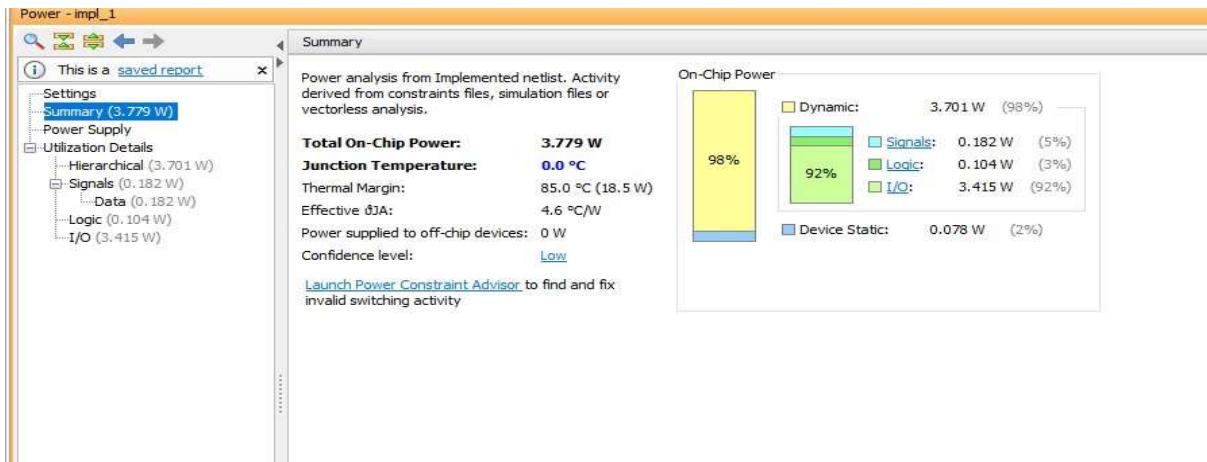


Figure 7.2.4 power report for 16-bit multiplier

The power analysis derived from the implemented netlist, reveals a total on-chip power consumption of 3.779 W. This power is predominantly dynamic in nature, accounting for 3.701 W (98%), while the remaining 0.078 W (2%) is attributed to static or leakage power. A detailed breakdown of the dynamic power indicates that the I/O components consume the majority, drawing 3.415 W, which constitutes approximately 92% of the dynamic power. Signal switching contributes 0.182 W (5%), and internal logic operations consume 0.104 W (3%). The dominance of I/O power

highlights significant switching activity at the interface level, which could stem from high toggle rates or strong drive strengths required for off-chip communication. The thermal profile shows a junction temperature baseline of 0.0°C, with a thermal margin of 85.0°C, and an effective thermal resistance (θ_{JA}) of 4.6°C/W.

In conclusion, by replacing the Han-Carlson Adder with the Kogge-Stone Adder has improved the overall speed due to faster carry propagation, as evident from reduced logic delays. However, this comes with increased power consumption, particularly I/O power, and higher net delays due to complex routing. The Kogge-Stone Adder is ideal for performance-critical designs, while the Han-Carlson Adder may still be preferred in power- or area-sensitive applications. Thus, the choice depends on the specific trade-offs between speed, power, and area.

7.3 Comparison of existing and proposed methods

Compressor Type	LUT	IBUF	OBUF	Cells	Dynamic power(W)	Static power(W)	Chip power(W)
3:2(Exact)	2	3	2	7	0.900	0.120	1.02
4:2(Exact)	3	5	3	11	1.545	0.135	1.68
5:3	3	5	3	11	1.483	0.078	1.562
15:4	8,4,8	15	4	39	3.701	0.078	3.779

Compressor Type	LUT	IBUF	OBUF	Cells	Dynamic power(W)	Static power(W)	Chip power(W)
3:2(Approx)	2	3	2	7	0.712	0.116	0.828
4:2(Approx)	2	4	2	8	0.995	0.122	1.11
5:3	3	5	3	11	1.483	0.078	1.562
15:4	8,4,8	15	4	39	3.701	0.078	3.779

Figure 7.3.1 Comparison of existing and proposed methods

7.4 CONCLUSION

The design and implementation of a low-power approximate multiplier using a 15:4 compressor have demonstrated a promising balance between performance efficiency and power reduction. By incorporating approximation techniques within the compression stage, the proposed design achieves substantial improvements in power consumption and area optimization while maintaining acceptable computational accuracy for error-tolerant applications. The use of the 15:4 compressor effectively reduces the critical path delay and logic complexity compared to traditional multiplication architectures, leading to faster and more energy-efficient operations. Simulation results confirm that this multiplier is well-suited for applications in image processing, machine learning, and IoT devices, where low power and high speed are prioritized over exact arithmetic precision. Overall, the proposed approach validates the potential of approximate computing in modern VLSI systems, offering a scalable and practical solution for energy-aware digital arithmetic units.

The proposed architecture is based on array structure. The design depends on the approximate compressors as well as adders. The speed of design mainly depends on the speed of compressors and the parallel prefix adder i.e., KSA. The total process is designed by using Verilog HDL and synthesized on Xilinx Vivado. The Output and RTL Schematics are observed on the tool. These dependencies carry important information about the structure of the objects in the visual scene. Luminance masking is a phenomenon whereby image distortions (in this context) tend to be less visible in bright regions, while contrast masking is a phenomenon whereby distortions become less visible where there is significant activity or "texture" in the image.

Entropy

The entropy of a system as defined by Shannon gives a measure of uncertainty about the images' actual structure. Shannon's function is based on the concept that the information gain from an event is inversely related to its probability of occurrence. Several authors have used Shannon's concept for image processing and pattern recognition problems. Many used Shannon's concept to define the entropy of an image assuming that an image is entirely represented by its gray level histogram only. As a result segmentation algorithms using Shannon's function resulted in an unappealing result, same entropy and threshold values for different images with identical histogram.

Shannon defined the entropy of ann-state systems as

$$H = - \sum_{i=1}^n p_i \log p_i$$

Where p_i is the probability of occurrence of the event I and

$$\sum_{i=1}^n p_i = 1, 0 \leq p_i \leq 1$$

REFERENCES:

1. Andrew, B, Kahng, Seokhyeong, Kang 2012, Accuracy configurable adder for approximate arithmetic designs, DAC Design Automation Conference, SanFrancisco, USA, pp. 820-825.
 2. Jinghang Liang, Jie Han & Fabrizio Lombardi 2013, New metrics for the reliability of approximate and probabilistic adders, IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760-1771.
 3. Vaibhav Gupta, Debabrata Mohapatra, Raghunathan & Kaushik Roy 2013, Low-power digital signal processing using approximate adders, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 1, pp. 124-137.
 4. M. Shafique, S. Garg, J. Henkel and D. Marculescu, "The EDA challenges in the dark silicon era," 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), 2014, pp. 1-6
 5. Pawan Sonware, Malathi, P & Manish sharma 2015, Design of low power inexact 4:2 compressor using approximate adder, International Conference on Computer Communication and Control, Indore, India,pp. 1-5.
 6. Jothin, R & Vasanthanayaki, C 2018, High performance error tolerant adders for image processing applications, IETE Journal of Research, vol. 67, no. 2, pp. 205-216.
 7. Ayad Dalloo, Ardalan Najafi & Alberto Garcia-Ortiz 2018, Systematic design of an approximate adder: the optimized lower part constant-or adder, IEEE Transactions on Very Large-Scale Integration Systems, vol. 26, no. 8, pp. 1595-1599.
 8. Gnanambikai Palanisamy, Vijeyakumar Krishnasamy Natarajan & Kalaiselvi Sundaram 2019, Area efficient parallel adder with faithful approximation for image and signal processing applications, IET Image Processing, vol. 13, no.13,pp. 2587 -2594.
-

9. Hsiao, SF, Jiang, MR & Yeh, JS 1998, Design of high-speed low- power 3–2 counter and 4–2 compressor for fast multipliers, Electronics Letters, vol. 34, no.4, pp. 341–343.
10. Chang C.-H. Gu J. and Zhang M. (2004) Ultra low-voltage low-power cmos 4-2 and 5-2 compressors for fast arithmetic circuits IEEE Transaction on Circuit and System I 51(10) 1985–1997.
11. Minho, ha & Sunggu, Lee 2018, Multipliers with approximate 4-2 compressors and error recovery modules, IEEE Embedded Systems Letters, vol. 10, no. 1, pp. 6-9.
12. Guo, Y, Guo, L, Kimura, S & Sun, H 2018, Low-cost approximate multiplier design using probability-driven inexact compressors, IEEE Asia Pacific Conference on Circuits and Systems, Chengdu, China, pp. 291– 294.
13. Yen-Jen, Chang, Yu-Cheng, Cheng, Yi-Fong, Lin, Shao-Chi, Liao, Chun-Hsiang, Lai & Tung-Chi, Wu 2019, Imprecise 4-2 compressor design used in image processing applications, IET Circuits Devices Systems, vol. 13, no.6, pp. 848-856.
14. Kenneth Steiglitz& Peter R. Cappello 1983, A VLSI layout for a pipelined dadda multiplier, ACM Transactions on Computer Systems, vol. 1, no. 2, pp. 157-174.
15. Ming-Roun Jiang 1998, Design of high-speed low-power 3-2 counter and 4-2 compressor for fast multiplication, Electronics Letters, vol. 34, no. 4, pp. 341-343.
16. Radhakrishnan, D & Preethy AP 2000, Low power CMOS pass logic 4- 2 compressor for high-speed multiplication, IEEE Midwest Symposium on Circuits and Systems, vol. 3, Lansing, MA, USA, pp. 1296-1298.
17. Jiungmin, Gu & Chip-Hong, Chang 2003, Ultra-low voltage, low power 4-2 compressor for high-speed multiplication, International Symposium on Circuits and Systems,

18. Dursun Baran, Mustafa Aktan & Vojin, G. Oklobdzija 2010, ‘Energy efficient implementation of parallel CMOS multipliers with improved compressors’, ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), USA.
19. Pourormazd, M, Bahram Rashidi & Bahman Rashidi 2011, ‘Design and implementation of low power digital FIR filter based on low power multipliers and adders on Xilinx FPGA’, International Conference on Electronics Computer Technology, vol. 2, Kanyakumari, India, pp. 18–22.
20. Momeni & Lombardi 2015, ‘Design and analysis of approximate compressors for multiplication’, IEEE Transactions on Computers, vol. 64, no. 4, pp. 984-994.
21. Omid Akbar 2017, ‘Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers’, IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 25, no. 4, pp. 1352-1361.
22. Karri Manikantta Reddy 2019, ‘Design and analysis of multiplier using approximate 4-2 compressor’, AEU International Journal of Electronics and Communication, vol. 107, pp. 89-97.
23. Zhongde Wang, G, A, Jullien and W, C, Miller 1995, ‘A new design technique for column compression multipliers’, IEEE Transactions on Computers, vol. 44, no.8, pp. 962-970.
24. Guoping Wang & James Shield 2005, ‘The efficient implementation of an array multiplier’, International Conference on Electro Information Technology, Lincoln, NE, USA, pp. 5.
25. Kyaw, KY, Goh, WL & Yeo, KS 2010, ‘Low-power high – speed multiplier for error-

tolerant applications‘, IEEE International Conference on Electron Devices and Solid-State Circuits, Hongkong, China, pp. 1-4.

26. Chia-Hao, Lin & Ing-Chao, Lin 2013, High accuracy approximate multiplier with error correction using inaccurate 4-2 counter‘, International Conference on Computer Design, Asheville, NC, USA, pp. 33-38.
 27. Pekmestzi, K, Soudris, D, Tsoumanis, K, Xydis, S & Zervakis, G 2015, Hybrid approximate multiplier architectures for improved power- accuracy trade-offs, IEEE/ACM International Symposium on Low Power Electronics and Design, Rome, Italy.
 28. Srinivasan Narayananamoorthy, Hadi Asghari Moghaddam, Zhenhong Liu, Taejoon Park & Nam Sung Kim 2015, Energy-efficient approximate multiplication for digital signal processing and classification applications‘, IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 23, no. 6, pp. 1180-1184.
 29. Kopparapu Manikanta, Manjunath, Sivanantham S, Sivasankaran K & Venama Harikiran 2015, Design and implementation of 16x16 modified booth multiplier‘, IEEE Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, pp. 1-5.
 30. Suganthi Venkatachalam, Hyuk Jae Lee & Seok-Bum Ko 2018, Power efficient approximate booth multiplier‘, IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy.
 31. IhsenAlouani, HamzehAhangari, Ozcan Ozturk & Smail Niar 2018, A novel heterogeneous approximate multiplier for low power and high performance‘, IEEE Embedded Systems Letters, vol. 10, no.2, pp. 45-48.
 32. AlokeSaha, Rahul Pal, B, Akhilesh, G, Naik & Dipankar Pal 2018, Novel CMOS multi-bit counter for speed-power optimization in multiplier design‘, AEU - International
-

33. VijeyaKumar, KN, Elango, S & Kalaiselvi, S 2018, ‘VLSI implementation of high-speed energy-efficient truncated multiplier’, Journal of Circuits, Systems and Computer, vol. 27, no.5, pp. 1850077.
34. Bhardwaj (2014) wallace tree multiplier for error-resilient systems in 1th International Symposium on Quality Electronic Design pp. 263–269.
35. Lau, M. S. K., Ling, K.-V., and Chu, Y.-C. (2009). Energy-aware probabilistic multiplier: design and analysis. In Proceedings of the international conference on Compilers, architecture, and synthesis for embedded systems (pp. 281-290).
36. Venkatesan, R., Agarwal, A., Roy, K., and Raghunathan, A. (2011). Macaco: Modeling and analysis of circuits for approximate computing. In Proceedings of international conference on Compilers, architecture, and synthesis for embedded systems (pp. 67-73).
37. Farshchi, F., Abrishami, M. S., and Fakhraie, S. M. (2013). New approximate multiplier for low power digital signal processing. In 8th Midwest Symposium on Circuits and Systems (pp. 25-30).
38. Jinghang Liang, Jie Han, and Fabrizio Lombardi. (2013). New metrics for the reliability of approximate and probabilistic adders. IEEE Transactions on Computers, 62(9), 1760-1771.
39. Parandeh-Afshar, H., Brisk, P., and Ienne, P. (2009). An FPGA logic cell and carry chain configurable as a 6:2 or 7:2 compressor. Technology and Systems, 2(3), 1-42.
40. Dandapat, A., Ghosal, S., Sarkar, P., and Mukhopadhyay, D. (2010). A 1.2-ns 16 x 16-bit binary multiplier using high-speed compressors. International Journal of Electrical and Electronic Engineering, 4(3), 485-490.
41. Rouholamini, M., Kavehie, O., Mirbaha, A. P., Jasbi, S. J., and Navi, K. (2007). A new design for 7:2 compressors. In IEEE/ACS International Conference on Computer Systems and Applications (pp. 474-478).
42. Veeramachaneni, S., Krishna, M., Avinash, L., Puppala, S. R., and Srinivas, M. B. (2007). Novel architectures for high-speed and low-power 3:2, 4:2, and 5:2 compressors.

In 20th International Conference on VLSI Design and 6th International Conference on
Embedded Systems (pp. 324-329).

