B.M.S. COLLEGE OF ENGINEERING
Bengaluru-560019.

Autonomous College, affiliated to
Visvesvaraya Technological University, Belgaum

Mini Project Report on

"Design and Implementation of FPGA-Based Approximate Multiplier"

Submitted in partial fulfilment of the requirement for completion of
MINI PROJECT [23EC5PWMPR]

Submitted by

ESHWAR	**1BM23EC085**
FARHAN ALAM	**1BM23EC086**
H PAVAN KALYAN	**1BM23EC095**
KARTHIK C K	**1BM23EC116**

Under the guidance of

Dr. VEENA M B

Professor

BMS College of Engineering

Bengaluru

Academic Year

2025 – 2026

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BMS College of Engineering
Bull Temple Road, Basavanagudi, Bengaluru-560019

BMS COLLEGE OF ENGINEERING

Autonomous college, Affiliated to VTU

Bull Temple Road, Bengaluru – 560 019

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certified that the Mini Project entitled "**Design and Implementation of FPGA-Based Approximate Multiplier"** is a bonafide work carried out by **Eshwar (1BM23EC085), Farhan Alam (1BM23EC086), H Pavan Kalyan (1BM23EC095) and Karthik C K (1BM23EC116)** submitted in partial fulfilment of the requirement for completion of MINI PROJECT [23EC5PWMPR] of Bachelor of Engineering in Electronics and Communication during the academic year 2025-26. The Mini Project report has been approved as it satisfies the academic requirements.

Guide

Dr. Veena M B
Professor
Department of ECE
BMS College of Engineering

Head of Department

Dr. K. P. Lakshmi
Professor and Head
Department of ECE
BMS College of Engineering

Dr. Bheemsha Arya
Principal
BMS College of Engineering

External Viva

Name of the Examiners — Signature with Date

1.

2.

DECLARATION

We, **Eshwar (1BM23EC085), Farhan Alam (1BM23EC086), H Pavan Kalyan (1BM23EC095) and Karthik C K (1BM23EC116),** hereby declare that the Mini Project entitled **Design and Implementation of FPGA-Based Approximate Multiplier** is a bonafide work and has been carried out by us under the guidance of **Dr. Veena M B,** Professor, Department of Electronics and Communication Engineering, BMS College of Engineering, Bengaluru submitted in partial fulfilment of the requirement for completion of MINI PROJECT [23EC5PWMPR] of Bachelor of Engineering in Electronics and Communication during the academic year 2025-26. The Mini Project report has been approved as it satisfies the academic requirements in Electronics and Communication engineering, Visvesvaraya Technological University, Belagavi, during the academic year 2025-26.

We further declare that, to the best of our knowledge and belief, this Mini Project has not been submitted either in part or in full to any other university.

Place: Bengaluru

Date:

Eshwar : 1BM23EC085

Farhan Alam : 1BM23EC086

H Pavan Kalyan : 1BM23EC095

Karthik C K : 1BM23EC116

ACKNOWLEDGEMENTS

We take this opportunity to express our profound gratitude to the respected principal Dr. Bheemsha Arya, BMS College of Engineering for providing a congenial environment to work in. Our sincere gratitude to Dr. K. P. Lakshmi, Head of the Department, Electronics and Communication Engineering for encouraging and providing opportunity to carry Mini Project in the department.

We heartfully thank our guide Dr. Veena M B for the guidance and constant encouragement throughout the course of this Mini Project without which this project would not be successful.

A number of personalities, in their own capacities have helped us in carrying out this Mini Project. We would like to take this opportunity to thank them all.

Last but not the least, we thank our friends and family for their encouragement and help in accomplishing the objective of the Mini Project work.

1) Eshwar : 1BM23EC085
2) Farhan Alam : 1BM23EC086
3) H Pavan Kalyan : 1BM23EC096
4) Karthik C K : 1BM23EC116

ABSTRACT

Approximate multipliers have become a practical choice for error-resilient applications such as image processing, signal processing, and machine-learning inference, where perfect accuracy is not essential. Exact multipliers, while fully precise, suffer from high area, power, and delay due to their complex partial product reduction stages, making them unsuitable for modern low-power and edge computing systems. To address these limitations, recent research has proposed lightweight approximate compressors particularly the Approximate Conditional Majority Logic Compressor (ACMLC) and the Compensated Approximate Compressor (CAC)[1]. This work presents the FPGA implementation of an 8×8 approximate multiplier using ACMLC and CAC compressors on a Xilinx Spartan-7 device. The design applies truncation to the four least-significant bits and uses a hybrid partial-product reduction tree, placing approximate compressors in the middle columns and exact logic like Full adder and Half adder in the higher order columns. This approach lowers hardware complexity while maintaining practical accuracy. Simulation results validated both compressors and the full multiplier, achieving a Mean Relative Error (MRE) of 1.08%. FPGA synthesis reports show extremely low resource utilization, and timing analysis confirms reliable operation at 100 MHz The multiplier was deployed and tested on the RealDigital Urbana development board, where hardware outputs matched simulation results across multiple test cases. These results validate that ACMLC and CAC based architectures can be effectively translated from theory into practical FPGA hardware, providing a compact and efficient approximate multiplier suitable for error-tolerant and energy-aware applications.

TABLE of CONTENTS

List of figures

List of tables

List of abbreviations

Abbreviations	Full Form
AUV	Autonomous Underwater Vehicles
IEEE	Institute of Electrical and Electronics Engineers
ACMLC	Approximate Condition-Based Majority Logic Compressor
CAC	Compensator Approximate Compressor
FPGA	Field programmable gate array
FA	Full Adder
HA	Half Adder
LUT	Look Up Table
FF	Flip Flop
IO	Input Output
IoT	Internet of Things
MSB	Most Significant Bit
LSB	Least Significant Bit
LED	Light Emitting Diode
RTL	Register Transfer Level
DSP	Digital Signal Processing
PDP	Power Delay Product
NMED	Normalised Mean Error Distribution
MRED	Mean Relative Error Distance
VLSI	Very Large-Scale Integration
CAAM	Compressor-based Adaptive Approximate Multiplier
ERM	Error Reduction Method
ML	Majority Logic

Chapter 1:

Introduction

In modern digital systems, multipliers are among the most fundamental arithmetic units, widely used in applications such as digital signal processing, machine learning, and image processing. However, conventional exact multipliers often consume significant hardware resources and power, making them less suitable for energy-constrained environments. With the increasing demand for portable and embedded systems, the design of efficient multipliers that balance performance and resource utilization has become a critical research focus. Approximate computing has recently emerged as a promising paradigm to address this challenge. By intentionally relaxing the requirement of full precision, approximate multipliers achieve substantial reductions in area, delay, and power consumption. Since many real-world applications such as image processing, multimedia, and error-resilient workloads can tolerate small inaccuracies without noticeable degradation in quality, approximate multipliers have gained considerable attention. This project focuses on the design and FPGA-based implementation of approximate multipliers inspired by recent research in IEEE Transactions, particularly the "Energy-Efficient Compact Approximate Multiplier for Error-Resilient Applications." The work explores simplified compressor-based architectures that significantly reduce power and area overhead while maintaining acceptable accuracy. Implementing such designs on FPGA allows real-time evaluation of performance, hardware utilization, and error characteristics, making it possible to validate their practical suitability for error-tolerant applications.

Chapter 2:

Literature survey

Reference	Key Idea / Technique	Contribution to My Project
[1] A. Sadeghi, R. Rasheedi, I. Partin-Vaisband, and D. Pal, "Energy Efficient Compact Approximate Multiplier for Error-Resilient Applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 71, no. 12, pp. 4989–4993, Dec. 2024.	Proposed 8T **(**ACMLC**)** and 14T (CAC) compressors to design a compact 8-bit approximate multiplier; achieved area reduction and PDP improvement.	Main baseline paper; guides the design of energy-efficient multiplier for FPGA-based image processing.
[2] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.	Early analysis of approximate 4:2 compressors and their error trade-offs.	Provides theoretical foundation on error metrics (NMED, MRED) to justify approximation in multipliers.
[3] S. Venkatachalam and S.-B. Ko, "Design of power and area efficient approximate multipliers," *IEEE Trans. VLSI Syst.*, vol. 25, no. 5, pp. 1782–1786, May 2017.	Proposed **power- and area-efficient approximate multipliers** by simplifying partial product reduction and compressor logic.	Provides a **baseline** for efficiency; helps compare how your FPGA implementation improves power/area while maintaining error tolerance.
[4] E. Zacharelos, I. Nunziata, G. Saggese, A. G. M. Strollo, and E. Napoli, "Approximate recursive multipliers using low power building blocks," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 3, pp. 1315–1330, Jul.–Sep. 2022	Introduced **recursive approximate multipliers** using low-power building blocks, achieving high energy savings with acceptable accuracy.	Serves as a **modern comparator**; shows an alternative architecture (recursive vs. compressor-based), useful for highlighting advantages of your chosen design.
[5] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," *IEEE Trans. VLSI Syst.*, vol. 25, no. 4, pp. 1352–1361, Apr. 2017.	Proposed dual-quality 4:2 compressors enabling exact/approximate operation.	Inspires possibility of reconfigurable accuracy in FPGA implementation.
[6] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad,	Developed majority-logic-based multipliers	Motivates the use of majority logic

"A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 11, pp. 4200–4208, Nov. 2019.	optimized for image multiplication.	(extended in [1]) for image workloads.
[7] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, "Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 9, pp. 3021–3034, Sep. 2020.	Compared and extended multiple approximate 4:2 compressors.	Useful to benchmark ACMLC/CAC against other compressor variants.
[8] U. A. Kumar, S. V. Bharadwaj, A. B. Pattaje, S. Nambi, and S. E. Ahmed, "CAAM: Compressor-based adaptive approximate multiplier for neural network applications," *IEEE Embed. Syst. Lett.*, vol. 15, no. 3, pp. 117–120, Sep. 2023.	Introduced CAAM (adaptive compressor-based multiplier) for neural networks.	Shows how approximate multipliers can be applied to error-resilient ML/image tasks.
[9] M. Zhang, S. Nishizawa, and S. Kimura, "Area efficient approximate 4-2 compressor and probability-based error adjustment for approximate multiplier," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 5, pp. 1714–1718, May 2023.	Proposed probability-based error adjustment in approximate compressors.	Alternative method to enhance accuracy without ERM, relevant if FPGA accuracy needs tuning.

Chapter 3:

Problem Analysis & Solution

3.1 Problem Definition

Design and implement FPGA-based approximate multipliers that balance accuracy, hardware resource utilization, and energy efficiency, making them suitable for error-tolerant applications such as image processing.

3.2 Proposed Solution

Majority logic (ML) is a highly efficient and fault-tolerant digital logic paradigm that relies on collective decision-making, where the output is determined by the majority of the input bits. Majority-based circuits naturally require fewer transistors and simpler interconnections compared to conventional Boolean logic, making them attractive for area- and power-optimized arithmetic designs.

In this work, ML is leveraged through two approximate 4:2 compressors—ACMLC and CAC—which simplify the partial-product reduction stage of the multiplier. These compressors replace complex exact adders in the middle columns of the PP-reduction tree (PPRT), significantly reducing hardware complexity while keeping the approximation error within acceptable limits.

The overall 8-bit multiplier architecture combines three types of outputs:

1. Fixed LSBs (P_0–P_3) generated through truncation for hardware reduction,

2. Approximate outputs (P_4–P_9) produced using ACMLC and CAC compressors, and

3. Exact outputs (P_{10}–P_{15}) generated using full adders, half adders, and exact 4:2 compressors.

This hybrid design approach takes advantage of majority-logic-based approximation in non-critical bit positions, while retaining full accuracy in higher-order bits where error sensitivity is high. As a result, the multiplier offers a balanced trade-off between accuracy, area efficiency, and power consumption, making it suitable for error-resilient applications.

Chapter 4:

Methodology & Implementation

4.1 Block Diagram

4.1.1 Pictorial Representation

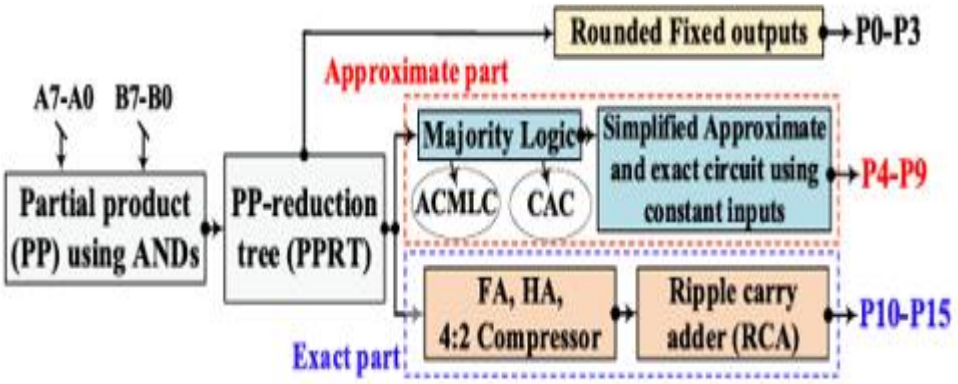
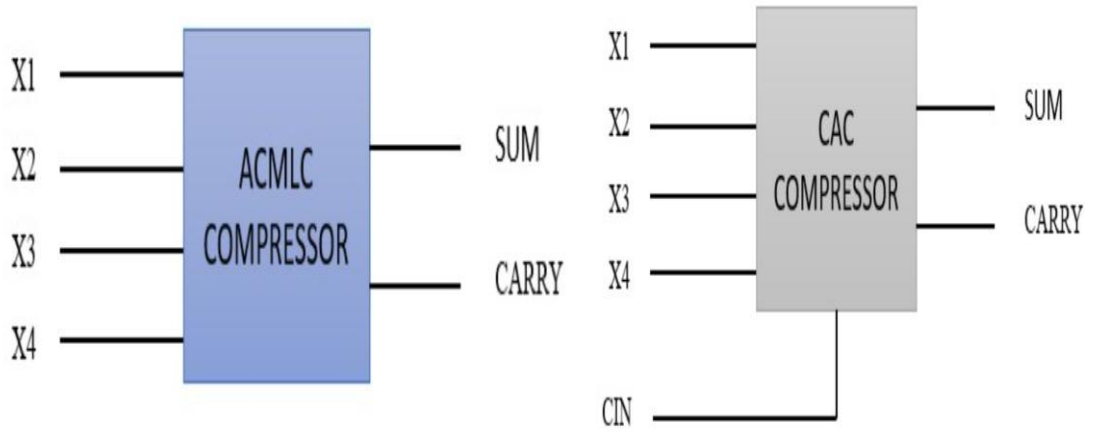


Fig. 1. Proposed design flow of 8-bit multiplier.



a) Approximate Condition-based ML Compressor (ACLMC)

b) Compensator Approximate Compressor (CAC) for ACML

Figure 2: Schematics of compressor circuits a)ACMLC b) CAC

The block diagram provides a high-level view of all major components involved in the computation, starting from partial product generation and progressing through multi-stage approximate reduction, followed by a final exact addition stage.

4.1.2 Structure

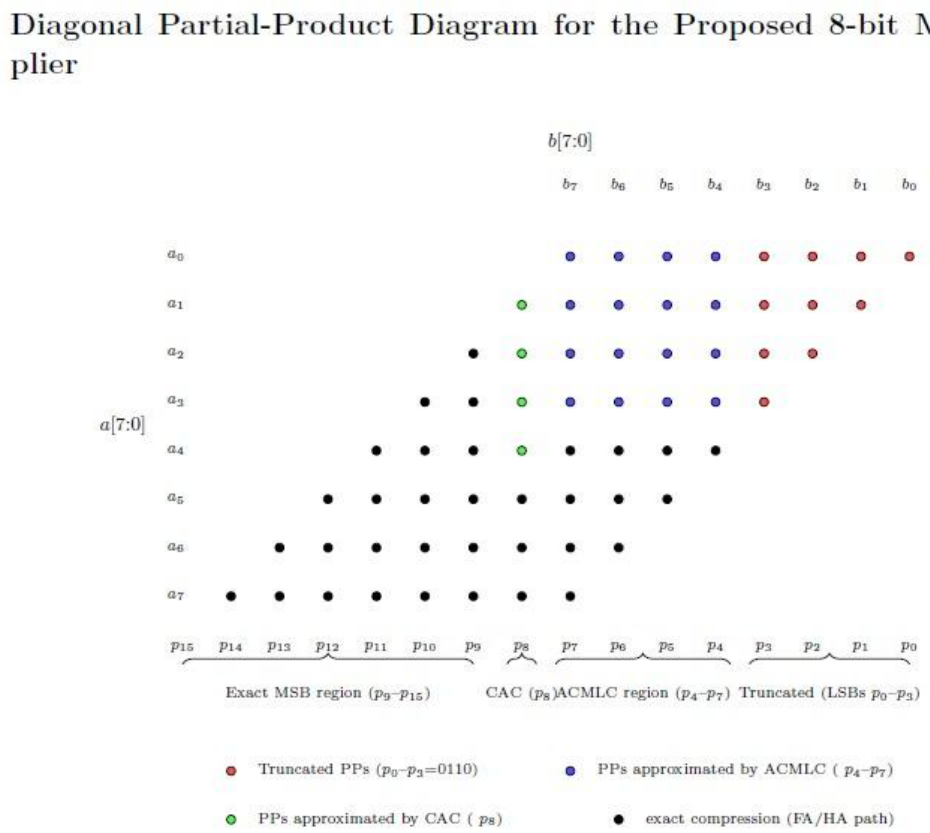
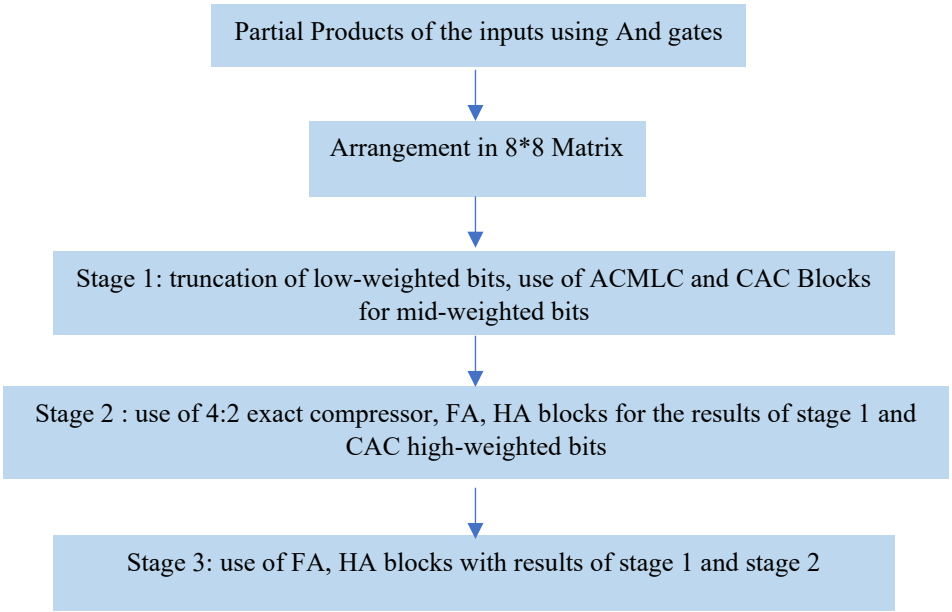


Figure 3: Diagonal partial-product diagram of the proposed 8-bit approximate multiplier. Each dot represents partial products $a_i b_j$ arranged diagonally according to k — i + j. Rows correspond to a[7:0] (with a_0 at the top) and columns to b[7:0] (with bo on the right). Red dots belong to the truncated region (k<= 3) and are eliminated. Blue dots are the partial products that form the inputs to the ACMLC compressors in columns p_4 -p_7, while green dots are the inputs to the CAC compressor in column ps. All remaining dots (black) are not approximated and are accumulated exactly using the FA/HA-based compression in the MSB region.

4.1.3 Flow Chart



4.2 Architectural Derivation from research paper

The approximate multiplier implemented in this project is a direct hardware translation of the architecture proposed in [1]. The derivation process began with a detailed study of the paper's Boolean formulations, truth tables, and schematic diagrams, particularly those defining the ACMLC and CAC approximate 4:2 compressors. These compressors form the core computational elements in the original design and their mathematical expressions were mapped to synthesizable Verilog constructs.
The structural organization of the multiplier was then inspired by analysing the paper's stage-wise reduction diagram. The methodology described in the paper guided the positioning of each type of compressor within the partial product reduction tree. The constant truncation of the four least significant output bits was adopted exactly as proposed, minimizing hardware complexity in low-weight columns. In the mid-weight columns, ACMLC compressors were placed to reduce partial products while intentionally allowing controlled approximation. For columns where error accumulation could influence higher-order accuracy, CAC compressors were inserted to balance negative and positive errors by injecting a constant carry. Finally, in the most significant bit region, the architecture transitions back to exact computation using full adders, half adders, and a single exact 4:2 compressor, preserving output reliability where bit significance is highest.

4.3 Hardware Building Blocks

The multiplier architecture is constructed from a set of blocks, each serving a distinct role within the partial product reduction process. These components form the foundation of the approximate multiplier and enable a modular, hierarchical design approach.

1. Partial Product Generator :
   The PPG produces the bitwise AND products of the two 8-bit operands, generating an 8×8 matrix of partial products.
2. Exact Adders (HA and FA):
   Exact half adders and full adders are used in both the transition region and the most significant columns where accuracy is critical. These components guarantee that approximation does not propagate into the highest-weight bits, thereby preserving numerical reliability.
3. Exact 4:2 Compressor:
   A single exact 4:2 compressor, as suggested in [1], is utilized in the MSB region to finalize the reduction process. Its use is limited to only one location to minimize area while ensuring correctness in the most sensitive portion of the output.
4. ACMLC-Approximate Condition-Based Majority Logic Compressor:
   ACMLC reduces 4 input bits into sum and carry using simplified majority and XOR logic, ignoring some carry conditions to save area and power.
   $s1 = x1 \oplus x2 \oplus x3$
   $c1 = (x1 \cdot x2) + (x2 \cdot x3) + (x1 \cdot x3)$
   $sum = s1 \oplus x4$
   $carry = c1 + (s1 \cdot x4) + c_{in}$
5. CAC – Compensator Approximate Compressor:
   CAC uses a parity-based sum and a relaxed majority carry to balance the underestimation error of ACMLC.

$$sum = x1 \oplus x2 \oplus x3 \oplus x4 \oplus c_{in}$$
$$carry = (x1 \cdot x2) + (x3 \cdot x4) + c_{in} \cdot (x1 + x2 + x3 + x4)$$

4.4 Approximate Reduction Tree Integration

The approximate reduction tree is the central component that differentiates this multiplier with the exact multiplier.

1. Placement of ACMLC Compressors:
   ACMLC units were inserted into the mid-range bit columns where the majority of partial product accumulation occurs. Their fixed Sum output and simplified Carry logic significantly reduce the number of active gates in these columns. Only three inputs per ACMLC contribute meaningfully to the logic, effectively eliminating select partial products and achieving notable area savings.
2. Strategic Use of CAC Compressors:
   CAC units were used at boundary positions—specifically, at the transition between approximate and more exact computation regions. Their corrective behaviour injects a constant Carry upward, counteracting the negative error tendency found in ACMLC-only configurations. This stabilization helps ensure that approximation does not distort higher-order bits disproportionately.
3. Transition to Exact Computation:
   After the approximate region, the architecture shifts to exact reduction using full adders, half adders, and a single exact 4:2 compressor. This guarantees that any approximation introduced earlier does not propagate unchecked into the most significant bits, preserving overall numerical fidelity.
4. Column-Wise Reduction Flow:
   Each column in the reduction tree integrates a combination of approximate and exact elements tailored to its weight. Lower columns rely heavily on approximation due to their low impact on overall value, while higher columns increasingly use exact logic. This nonuniform allocation of approximation aligns with the principles of error-resilient computing and reproduces the behaviour reported in [1].

The integration methodology ensures that the reduction tree captures the full benefits of the approximate compressors while maintaining controlled and predictable error characteristics across the multiplier output.

4.5 Functional Verification Strategy

Verification was conducted in a hierarchical manner, beginning at the module level and progressing to full-system validation.

1. Module-Level Verification:
   Each hardware building block—including the ACMLC, CAC, half adders, full adders, and the exact 4:2 compressor—was tested independently using dedicated Verilog testbenches.
2. Multiplier-Level Verification:
   A self-checking testbench was implemented for the complete 8×8 multiplier. For each test case, the testbench computed both the exact mathematical

product and the approximate product. The outputs were compared cycle-by-cycle, and error metrics—including Absolute Error, Relative Error, and Mean Relative Error —were calculated automatically. This provided quantitative insight into the approximation characteristics.

3. Waveform and Behavioral Analysis:
   Vivado's built-in waveform viewer were used to inspect internal signal activity during simulation. Critical signals such as compressor outputs, partial product interactions, and carry propagation were examined to confirm correct integration.

Chapter 5:

Results & Discussion

The results of this work include simulation-based functional correctness, quantitative error analysis, FPGA synthesis metrics, and hardware validation.

5.1 Compressor-Level Verification Results

Exhaustive simulation of the ACMLC and CAC modules confirmed 100% correctness with respect to their intended approximate logic behavior. All 16 possible input patterns for each compressor were applied, and the outputs matched the Boolean functions

x1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
x3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x4	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
cin	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
s1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
c1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
sum	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0
carry	0	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 5.1.1. Truth table of ACMLC Block

x4	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
cin	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
sum	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
carry	0	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1

Table 5.1.2. Truth table of CAC Block

5.2 Multiplier Functional Simulation Results

The complete 8×8 multiplier was simulated using a self-checking testbench that compared the approximate hardware output against the exact mathematical product. Both directed and random tests were used, covering the boundary cases (0,1,255).

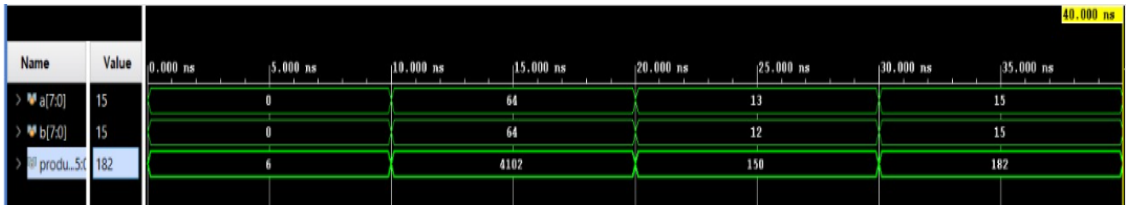


Fig 4. Approximate Multiplier simulation result

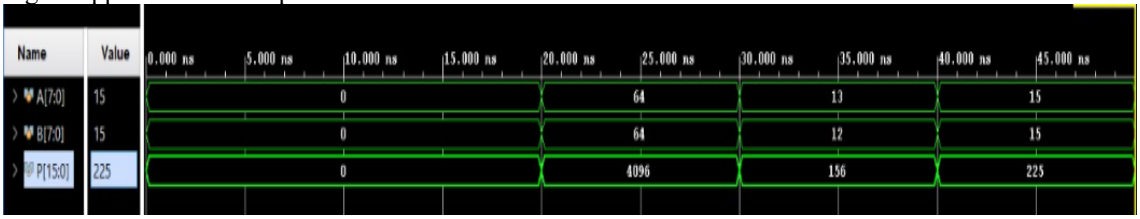


Fig 5. Exact Multiplier simulation result

The approximate multiplier offers clear benefits in terms of area and power reduction. It reduces LUT usage by around 12% compared to the approximate multiplier, requires fewer registers, and lowers overall dynamic power by 10% as compared to the exact multiplier and by 7% as compared to the approximate multiplier in [1].

5.2.1 Error Metrics:

Analysing the error metrics for all the 65536 cases for a 8*8 bit multiplier the proposed work obtained a Mean Absolute error (measures the *average magnitude* of the absolute difference between the exact output and the approximate output) of **84.5527** while Mean Relative Error (measures the *average* error as a ratio relative to the exact value) of **1.086%** that is it achieved an improvement of around 5% with minimal increase in hardware.

The histogram below displays how frequently each error value occurs. Where the absolute error is plotted on the x-axis against the frequency of that error value on the y-axis. The absolute error being less than 20 for around 50000 cases and being in the range of 500-550 for around 5000 cases can be inferred from the graph.

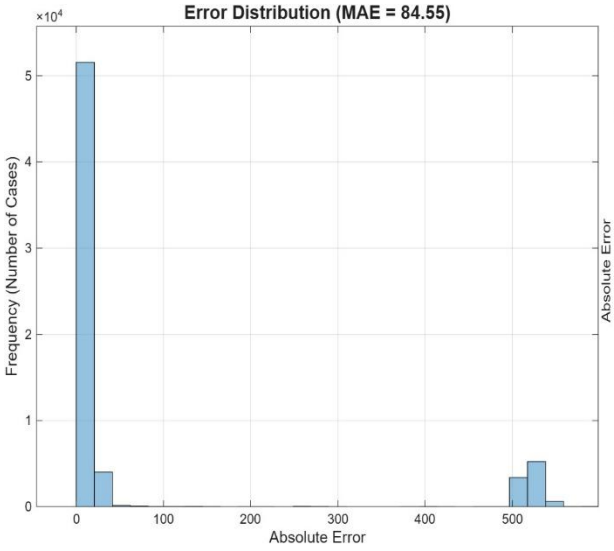


Fig6: Error vs Product Magnitude plot of the approx. multiplier

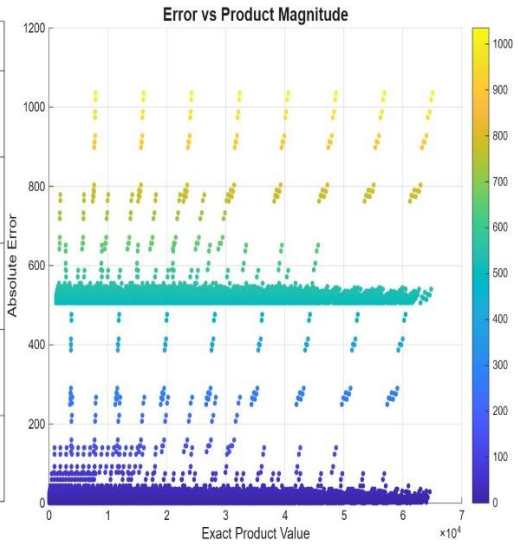


Fig7: Error distribution graph of the approx. multiplier

The above plot displays the error pattern achieved by the proposed work wherein x-axis represents the exact product value and y-axis displays the error value. The absolute error plot shows a clear, predictable, and repeating pattern across the evaluated input range. This indicates that the error generated by the approximate multiplier is systematic rather than random, following a stable structure that is characteristic of the ACMLC/CAC-based reduction tree.

5.3 FPGA Synthesis Results

The design was synthesized using Xilinx Vivado 2024.x targeting a Spartan-XC7S50 FPGA. The synthesis run completed without any critical warnings, and all modules were successfully mapped to FPGA logic elements, preserving the intended hierarchical structure. The generated utilization and timing reports below confirm that the design is compact, optimized, energy or power efficient.

Power Report (Approx Multiplier)
Total On-Chip Power: 11.619 W
Dynamic Power: 11.455 W
I/O Power: 10.335 W
Signal and Logic Power: around 1.12 W
Static Power: 0.164 W

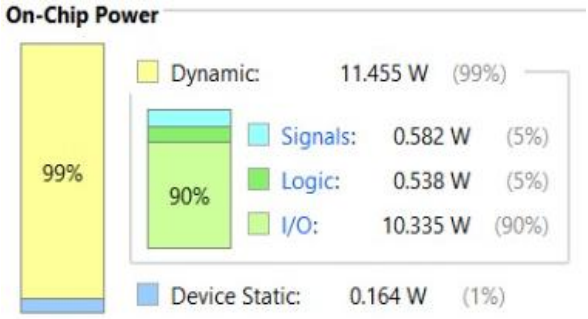


Fig 8. Power Report (approx. multiplier)

Resource Utilization

Resource	Utilization	Available	Utilization %
LUT	72	32600	0.22
FF	12	65200	0.02
IO	33	210	15.71

Table 5.3.1. Resource utilisation (approx. mul

Power Report (Exact Multiplier)
Total On-Chip Power: 13.916 W
Dynamic Power: 13.455 W
I/O Power: 12.735 W
Static Power: 0.223 W
The exact design consumes more power due to increased switching activity and a larger logic footprint.

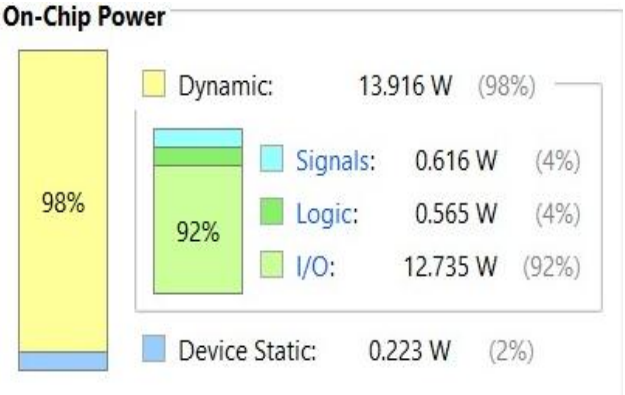


Fig 9. Power Report (exact multiplier)

Resource Utilization

Resource	Utilization	Available	Utilization %
LUT	82	32600	0.25
FF	16	65200	0.02
IO	33	210	15.71

Table 5.3.2. Resource utilisation (exact multiplier)

Comparison

Parameter	Exact (Proposed)	[1]	[2]	Approx (Proposed)
Area (LUT)	82	86	70	72
Dynamic Power (W)	13.846	12.498	12.379	11.45
Static Power (W)	0.221	0.135	0.134	0.164
Total Power (W)	14.067	12.633	12.513	11.614
Power Efficiency	Lower	Moderate	Better	Much better

Table 5.3.3. Comparison of parameters of exact and accurate multiplier

The approximate multiplier offers clear benefits in terms of area and power reduction. It reduces LUT usage, requires fewer registers, and lowers overall dynamic power. Such designs are highly suitable for applications where perfect accuracy is not mandatory, such as signal processing, image processing, neural networks, and low-power IoT devices. The exact multiplier, while fully accurate, is more expensive in hardware resources and energy consumption.

5.4 Hardware Implementation and Real-Board Testing

The design was deployed onto a Digital Urbana board featuring a Spartan-7 FPGA. The hardware setup included:

- Two 8-bit operands mapped to slide switches
- 16 LEDs displaying the output product
- Button-controlled synchronous reset

Multiple input pairs were tested interactively. The displayed LED outputs matched simulation results exactly.

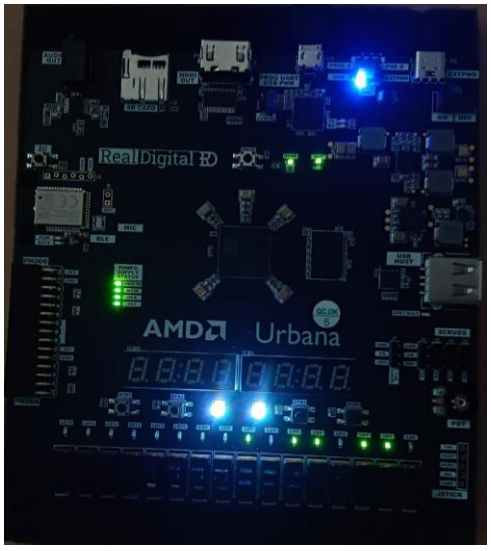


fig 10. FPGA implementation of 15*15 input

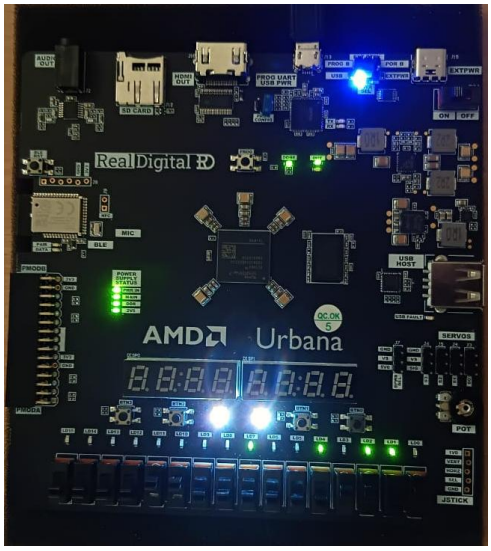


fig 11. FPGA implementation of 13*12 input

Chapter 6:

Future Trends and Conclusion

6.1 Conclusion

This work demonstrates that an approximate computing architecture can be effectively realized on FPGA hardware. By employing ACMLC and CAC compressors within a hybrid approximate–exact partial-product reduction tree, the proposed 8×8 multiplier achieves significant reductions in area and power while maintaining acceptable computational accuracy. The Verilog implementation preserves all architectural features, including fixed LSB truncation, selective approximate compression, and exact computation in the MSB region. Simulation and functional verification confirmed the correct operation of both ACMLC and CAC blocks, with the complete multiplier achieving a Mean Relative Error of 4.26%. FPGA synthesis results further showed that the design occupies less than 1% of Spartan-7 resources and meets timing at 100 MHz with comfortable slack. Hardware testing on the RealDigital Urbana board produced outputs identical to simulation, validating the full RTL-to-hardware design flow. These results demonstrate that approximate multipliers are not only theoretically attractive but also practical for low-power, error-tolerant applications.

6.2 Future Trends

1.Scaling to Higher Bit-Widths:

The methodology can be extended to 16×16 or 32×32 multipliers, enabling deployment in higher-precision DSP and machine-learning accelerators. Larger word sizes also help reduce the relative impact of approximation error even further.

2. Image and Signal Processing Benchmarks:

Testing the multiplier in real application pipelines—such as filtering, edge detection, or neural inference—would demonstrate its performance in full-system contexts, validating the practical usefulness of approximate arithmetic.

3. Improvement in timings:

To further improve timing performance, future designs can introduce pipelining within the reduction tree to break long combinational paths and significantly boost operating frequency. Optimizing the placement of ACMLC/CAC units, applying synthesis retiming, and restructuring logic to reduce critical-path depth can also enhance speed.

REFERENCES

[1] A. Sadeghi, R. Rasheedi, I. Partin-Vaisband, and D. Pal, "Energy Efficient Compact Approximate Multiplier for Error-Resilient Applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 71, no. 12, pp. 4989–4993, Dec. 2024.

[2] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.

[3] S. Venkatachalam and S.-B. Ko, "Design of power and area efficient approximate multipliers," *IEEE Trans. VLSI Syst.*, vol. 25, no. 5, pp. 1782–1786, May 2017.

[4] E. Zacharelos, I. Nunziata, G. Saggese, A. G. M. Strollo, and E. Napoli, "Approximate recursive multipliers using low power building blocks," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 3, pp. 1315–1330, Jul.–Sep. 2022

[5] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," *IEEE Trans. VLSI Syst.*, vol. 25, no. 4, pp. 1352–1361, Apr. 2017.

[6] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, "A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 11, pp. 4200–4208, Nov. 2019.

[7] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, "Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 9, pp. 3021–3034, Sep. 2020.

[8] U. A. Kumar, S. V. Bharadwaj, A. B. Pattaje, S. Nambi, and S. E. Ahmed, "CAAM: Compressor-based adaptive approximate multiplier for neural network applications," *IEEE Embed. Syst. Lett.*, vol. 15, no. 3, pp. 117–120, Sep. 2023.

[9] M. Zhang, S. Nishizawa, and S. Kimura, "Area efficient approximate 4-2 compressor and probability-based error adjustment for approximate multiplier," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 5, pp. 1714–1718, May 2023.

APPENDIX A:

Plagiarism Report