International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 1, 2026

Research Paper

NOVEL TUNABLE APPROXIMATE COMPRESSOR DESIGNS FOR ENERGY CONSTRAINED MULTIPLY-ACCUMULATE OPERATIONS IN EDGE AI

Guvvada Naresh [1], B.Vijaya Lakshmi [2]

[1] *Student, Department of ECE , Gonna Institute of Information Technology & Sciences, Anakapalli, Andhra Pradesh, India.*

[2] *Assistant Professor, Department of ECE , Gonna Institute of Information Technology & Sciences, Anakapalli, Andhra Pradesh, India.*

ABSTRACT

Energy-constrained or self-powered hardware platforms are becoming more and more important for edge artificial intelligence (AI) systems, which require extremely efficient compute units without sacrificing functional precision. Modern AI workloads primarily use multiply-and-accumulate (MAC) processes, which means optimizing them is essential for sustainable edge intelligence. In this work, new tunable approximation compressor designs are presented that allow MAC topologies to dynamically change accuracy-energy trade-offs. The suggested compressors include tenability knobs and low-complexity approximation techniques that enable real-time configuration according to workload precision requirements or available energy budgets. When included into MAC units, the designs maintain a satisfactory level of accuracy for AI applications that require error resilience while achieving significant reductions in power consumption, delay, and silicon area. The designs are proven to be efficient and feasible through hardware implementation and assessment using Xilinx Vivado, proving their applicability for energy-adaptive computation in next-generation edge AI and autonomous sensing systems.

Keywords: Tunable Approximation, Compressor Design, Edge AI, Multiply-Accumulate Units, Low-Power Hardware, Energy-Constrained Systems, Vivado Implementation, Approximate Computing

Received: 21-11-2025 Accepted: 28-12-2025 Published: 07-01-2026

1. INTRODUCTION

1.1 History and Purpose

The quick development of Edge Artificial Intelligence (Edge AI) has made it possible to analyze intelligent data directly on edge devices with little resources, like wearable technology, mobile platforms, autonomous sensors, and Internet of Things nodes. In contrast to cloud-based systems, edge devices must provide real-time performance under stringent power, size, and temperature constraints. As a result, creating hardware designs that are energy-efficient has become essential to implementing AI workloads at the edge.

As Edge Artificial Intelligence (Edge AI) spreads quickly, it has completely changed the way we use technology by shifting intelligence from centralized clouds to local devices like wearables, smartphones, and Internet of Things sensors. Deep neural networks (DNNs) are computationally demanding models, and edge devices are subject to stringent power, size, and thermal limitations, making their deployment at the edge extremely difficult. Multiply-accumulate (MAC) operations, which dominate execution time and energy usage in convolutional neural networks (CNNs), deep neural networks (DNNs), and filtering applications, are fundamental to the majority of AI and signal-processing workloads. An adder and a multiplier make up the majority of a MAC unit, with the multiplier requiring the most processing. For edge platforms, traditional precise arithmetic implementations become more inefficient as model complexity and data precision rise.

1.2 Using Edge AI for Approximate Computing

By taking advantage of the intrinsic mistake tolerance of AI applications, approximate computing has become a promising paradigm to get around the drawbacks of exact arithmetic. Small numerical errors can be tolerated by many inference-based workloads without having a substantial impact on system-level accuracy or perceptual quality. A significant reduction in power consumption, silicon area, and propagation latency is achieved by approximate arithmetic circuits by easing stringent correctness criteria. Approximate adders and multipliers are among the many arithmetic blocks that have been thoroughly examined. A significant amount of switching activity and logic depth is specifically caused by partial product reduction in multipliers. Thus, optimizing this phase can result in notable increases in energy efficiency.

1.3 Compressors' Function in Multiplier Architectures

Compressors, like Wallace and Dadda multipliers, are essential components of partial product reduction trees. They directly affect the delay, power, and area of the multiplier by converting several input bits into fewer output bits. Although they have a high logic complexity and propagation overhead, conventional precise compressors guarantee accurate arithmetic output. In order to solve this problem, approximation compressors have been suggested, which reduce internal logic by changing or doing away with carry-generation paths. Although these systems contain controlled computational mistakes, they provide considerable savings in power and area. The majority of current approximation compressors, however, are static in nature and provide a predetermined accuracy–energy trade-off that is decided upon during design.

1.4 Tunable approximation compressors are required.

Edge AI systems frequently function with fluctuating workloads and precision demands. For instance, higher approximation may be tolerated by low-power inference modes, but better precision may be necessary during crucial decision phases. Static approximation designs have limited practical usefulness in real-world edge contexts since they are unable to adjust to such fluctuations. By using mode-select or control signals, tunable approximate compressors have been proposed as a solution to this issue, allowing runtime control over approximation levels. Systems with such designs are ideal for energy-constrained Edge AI accelerators because they provide a dynamic balance between computing precision and energy efficiency.

1.5 Present Work's Scope

The design and analysis of new tunable approximate compressor designs suited for multiply-accumulate operations in Edge AI applications are the main topics of this thesis. Power consumption, delay, area, and error characteristics are assessed by integrating the suggested compressors into MAC units. The goal is to save a lot of energy without sacrificing sufficient accuracy for AI tasks.

1.6. Statement of Problem

Despite a great deal of study on approximation arithmetic circuits, the following drawbacks plague current approximate compressor designs: Most solutions provide predetermined degrees of approximation, which limits flexibility to accommodate different workloads and levels of precision. The use of static approximation results in either poor energy savings or needless loss of accuracy. little attention paid to MAC-level integration, especially for Edge AI accelerators. Runtime tenability and its effects on power-accuracy trade-offs are not well explored.

1.7. Objectives

1. Creating innovative tunable approximate compressor architectures that can function at various accuracy levels is the main goal of this research.
2. To incorporate the suggested compressors into Edge AI accelerator-compatible MAC modules.
3. To use common VLSI metrics for the analysis and optimization of the suggested designs' power, latency, and area.
4. To assess the accuracy of computations by utilizing error metrics including error rate, mean error, and error distance.
5. To assess the planned designs' accuracy and energy efficiency in comparison to current exact and approximate compressors.
6. To show that configurable approximations are appropriate for Edge AI applications with limited energy.

2. LITERATURE SURVEY

Wu, Y., et al. (2023) provided a thorough analysis of approximation multiplier designs with the goal of enhancing energy efficiency in contemporary computer systems. The authors examined the effects of approximation strategies on power, delay, and accuracy and categorized them according to algorithmic, architectural, and circuit-level changes. For energy-constrained applications like IoT and Edge AI systems, the study emphasized the significance of compressor optimization and partial product reduction.
In their comprehensive review of approximate arithmetic circuits, Jiang, H., et al. (2016) included adders, multipliers, and accumulators. The study carefully categorized approximation methods and assessed the trade-offs between them in terms of performance, error metrics, and energy usage. This work encouraged the use of approximation computing in error-tolerant applications by laying a solid theoretical framework for it.
In 2015, Momeni, A., Han, J., Montuschi, P., and Lombardi, F. suggested and examined a number of approximation compressor designs for multiplication. By evaluating compressor topologies using error distance, power, and delay metrics, the authors showed that compressor-level approximation dramatically lowers energy consumption. The approximate compressor-based multiplier design relies heavily on this work.
Two-quality 4:2 compressors with adjustable accuracy modes were presented by Akbari, O., et al. (2017). Adaptive trade-offs between energy and accuracy are made possible by the suggested systems' ability to switch between accurate and approximate functioning at runtime. In this paper, the idea of configurable approximation compressors for dynamic workloads is directly motivated.
The Rounding-Based Approximate (RoBA) multiplier was proposed by Zendegani, R., et al. (2016). It eliminates less significant partial products to reduce hardware complexity. The design maintained a respectable level of precision for signal processing applications while achieving notable energy savings. The work showed how well controlled approximation works for workloads that require a lot of MAC power.
In their 2015 study, Narayanamoorthy et al. introduced energy-efficient approximate multipliers for tasks including digital signal processing and categorization. With little effect on application-level precision, the authors demonstrated that approximation arithmetic dramatically lowers power usage. According to their findings, approximate multipliers are appropriate for MAC operations in AI systems.

In 2004, Chang, C.-H., Gu, J., and Zhang, M. suggested low-voltage 4–2 and 5–2 compressor designs that were tailored for high-speed arithmetic units. Even though this work was precise, it set the stage for subsequent approximation compressor research and offered valuable insights into compressor optimization for low-power systems.
The approximate 4:2 compressors for probabilistic multipliers that Nanammal, T., and Gangrade, J. (2022) devised were aimed at MAC-based signal processing. While

preserving acceptable error characteristics, the suggested compressors were able to reduce area and power consumption. In this study, the usefulness of compressor approximation in MAC architectures was illustrated.
A simplified carry-generation logic was used to suggest an efficient approximate 4:2 compressor in [IEIE SPC Journal] (2021). Improvements in power, latency, and area were assessed once the design was included into a multiplier architecture. The findings demonstrated significant energy savings when compared to traditional compressors.
Kumar, A., et al. (2020) evaluated twelve distinct 4:2 compressors in a comparison study of several approximation compressor designs. By analyzing trade-offs between power, delay, and error metrics, the study offered helpful recommendations for compressor selection in approximation multiplier designs.
A modified 4:2 compressor-based area-efficient approximation multiplier was proposed by Patel, R., et al. (2019). The design showed better silicon usage and lower power consumption, which made it appropriate for edge computing and embedded applications.
Singh, P., et al. (2019) used simplified full adders with approximation compressors to investigate multipliers that are both area- and power-efficient. While maintaining a reasonable level of computational precision for DSP applications, the suggested architecture reduced energy consumption.The De Gruyter Survey (2021) examined current developments in neural network accelerators and MAC units in approximation arithmetic circuits. The survey underlined the significance of designing with error awareness and pointed out unresolved issues with adaptive and adjustable approximation methods.
In their 2022 study, Shirkavand and Moaiyeri presented an accuracy-efficient approximate multiplier that utilized error compensating approaches. Through a clever combination of approximation compressors, the system maintained low power usage while reducing overall inaccuracy. This work focuses on how to strike a balance between accuracy and approximation.
Vahdat, S., et al. (2023) combined compressors with complementing error behaviors and examined compressor error features to increase approximation multiplier accuracy. Mean error distance was greatly decreased using this method without compromising energy efficiency.
For edge deep learning accelerators, Kim, J., et al. (2021) introduced a quantization-aware approximate multiplier. The design proved that low-precision neural networks and approximate arithmetic can be used together to yield considerable energy savings with very little loss of accuracy.
In 2024, Kumar et al. suggested an approximate multiplier based on NCFETs that is tuned for ultra-low power operation. The study demonstrated how new transistor technologies improve the advantages of approximation compressor-based systems.
Energy-efficient approximation compressor topologies utilizing CNTFET technology were presented by Rahimi, A., et al. (2025). The suggested compressors showed enhanced energy efficiency and respectable output quality when included into Dadda multipliers and tested for image processing applications.
An algorithm for layer-wise approximation for deep neural network accelerators, ALWANN, was proposed by Yazdanbakhsh, A., et al. (2019). Approximate multipliers can be used selectively across network levels without retraining, according to the study, which greatly lowers energy consumption.
In their comparative analysis of approximation multipliers, Mittal, S. (2018) examined application applicability, design methodologies, and error measures. The study highlighted important research gaps in adaptive and adjustable approximation and offered a thorough evaluation framework.

3. EXISTING SYSTEM

The majority of the multiply-and-accumulate (MAC) units utilized in edge AI accelerators in the current system are made with exact

arithmetic circuits. To achieve complete numerical accuracy, these systems use traditional compressor topologies (such 3:2, 4:2, and higher-order exact compressors) incorporated into Wallace tree or Dadda tree multiplier architectures. Correctness and determinism are the main design objectives of these MAC units, which makes them appropriate for workloads in conventional computing and general-purpose digital signal processing.

These particular MAC-based systems, however, are static and non-adaptive. They function with a set accuracy and supply voltage, independent of energy availability or workload sensitivity. Their higher propagation latency, high power consumption, and greater silicon area limit their applicability for energy-constrained edge AI platforms like wearables, autonomous sensors, and battery-powered IoT nodes.

Utilizing technology-level optimizations such operand isolation, voltage scaling, clock gating, and power gating, the current systems occasionally use them to reduce power problems. The mistake resistance built into AI workloads is not taken advantage of by these techniques, despite the fact that they somewhat lower dynamic and leaky power. Reducing the precision requirements would allow for additional energy savings, but the arithmetic logic stays correct.
the characteristics of the current system are:

- MAC devices with precise compressors and fixed accuracy
- Computation that is neither adjustable nor flexible
  Excessive area overhead, delay, and energy
- Limited applicability for AI applications requiring ultra-low power

The need for adjustable approximation compressor designs that can dynamically balance accuracy and energy efficiency is driven by these constraints, as your work suggests.

4. PROPOSED SYSTEM

The suggested adjustable 2:1, 3:2, 4:2, and 5:3 approximation compressors that can alternate between two approximate modes (AM-1 mode & AM-2 mode) are implemented. By altering the critical route, the suggested tunable compressor circuit incorporates two logic functions and offers extra functionality for turning on or off particular logic blocks in response to a control signal, enabling switching between several functional modes. The implementation of the suggested tunable compressor circuits at 2:1 (PTAxC21), 3:2 (PTAxC32), 4:2 (PTAxC42), and 5:3 (PTAxC53) is covered in the next subsection.

$$= \begin{cases} O_0^{t-21} = p_0 + p_1, & when\ S = 1 \\ O_0^{t-21} = p_0, & when\ S = 0 \end{cases}$$

We incorporate two logic functions that correspond to two distinct approximation levels in order to enable real-time switching in the proposed tunable 2:1 compressor (PTAxC21). When the suggested 2:1 compressor is realized, the first logic function is equivalent to the intermediate output logic function (O 21 int0 = p0 + p1) that was determined following the probability analysis (see Section III, Subsection B). The optimized logic output (O 21 0 = p0) acquired during hardware optimization is represented by the other logic function. The proposed PTAxC21 works similarly to the PAxC21 compressor circuit, producing a single output O t−21 0. The circuit functions in two modes (AM-1 and AM-2) that can be dynamically switched using a control signal S, as seen in Fig. 13 (a). In order to implement the suggested circuit (PTAxC21), we create an integrated logic path for the output O t−21 0. Depending on the mode of operation, different logic outputs can be chosen by using the control signal S to turn on or off specific logic blocks. As seen in Fig. 13 (b), when the control signal is high (S = '1'), Blocks 1 and 2 are turned on and transistor MN3 is turned off to create an output O t−21 0 = p0 + p1. Conversely, when the control signal is low (S = '0'), MN3 is activated and the other transistors are gated, resulting in a

International Journal of Engineering Research and Science & Technology

simplified logic expression at the output, O t−32 0 = p0

$$= \begin{cases} O_0^{t-32} = p_0 + p_1 \\ O_1^{t-32} = p_0 p_1 + p_2 \end{cases}, \quad when\ S = 1$$
$$\begin{cases} O_0^{t-32} = p_0 + p_1 \\ O_1^{t-32} = p_2 \end{cases}, \quad when\ S = 0$$

The logic function corresponding to the intermediate output logic function (O 32 int1 = p0p1 + p2) obtained after probability analysis and the optimized logic output obtained after hardware optimization (O 32 1 = p2) are integrated in the proposed tunable 3:2 compressor (PTAxC32) during its realization (see Section III, Subsection B). Although the suggested tunable 3:2 compressor generates two outputs, O t−32 0 and O t−32 1, only the critical route associated with O t−32 1 is altered to add flexibility. In order to provide an integrated logic path for output O t−32 1, we utilize the common logic function block (p2) in order to implement the suggested PTAxC32 circuit. As seen in Fig. 14 (b), the proposed PTAxC32 compressor operates as follows: when the control signal is high (S = '1'), both blocks (Block-1 & Block-2) are turned on, producing output O t−32 1 = p0p1 + p2. For logic p0p1 to remain at intermediate node X, the transistor MN10 is switched off. Conversely, Block-1 is turned off when the control signal is low (S = '0'), resulting in an output with a reduced logic expression, O t−32 1 = p2. This is done by gating the transistors' power supply (MP1-MP3, MN1-MN3), as shown in the design in Fig. 14 (c). To help avoid any metastable states, transistor MN10 is also set to logic "0" for the floating node (X). (14), which presents the final expression for the suggested tunable 3:2 compressor output.

$$= \begin{cases} O_0^{t-42} = (p_0 p_1) + (p_2 + p_3) \\ O_1^{t-42} = (p_2 p_3) + (p_0 + p_1) \end{cases}, \quad when\ S = 1$$
$$\begin{cases} O_0^{t-42} = (p_0 p_1) + (p_2 + p_3) \\ O_1^{t-42} = (p_0 + p_1) \end{cases}, \quad when\ S = 0$$

MAC unit and approximate multiplier One basic mathematical operation used to calculate a DNN's MAC operation is multiplication. However, the high latency, high power consumption, and substantial space overhead of conventional multipliers pose design problems, particularly when used in low-power edge devices. Consequently, we suggest creating an approximation multiplier (PAxC MUL) that enhances latency, resource usage, and energy efficiency in order to remedy this issue. We also introduce a customizable approximation multiplier (PTAxC MUL) implementation. To construct an approximation MAC unit, the designs of fixed and tunable multipliers are also used.

Approximate Multiplier Of Mac Design:

The suggested design approach for the approximate multiplier allows the hardware to be customized to satisfy the requirements of a particular application in terms of accuracy, energy efficiency, and area overhead. A partial product matrix (PPM) of height N and length 2N-1 is formed by the N×N multiplier, which yields N partial products. The multiplier's partial product reduction stage, which is a major performance bottleneck because of the enormous adder trees, is where the approximation is introduced (see Section II). Combining the suggested approximate compressors lowers the PPM. First, the partial product matrix is split into three sections: • The Approximate Section is made up of the bottom PPM columns (Col0-ColP−1), where P is the number of columns that need to be approximated (P ϵ N). • The Hybrid Section, which is made up of the PPM middle columns (ColP-ColP+K), where K is the number of hybrid columns that are present after approximated columns. • The exact section with PPM's upper columns (ColP+K+1–Col2N−1)

International Journal of Engineering Research and Science & Technology

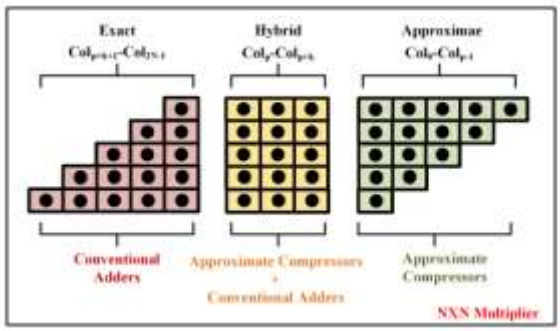

Fig.1. -Exact, Hybrid, and Approximate sections of a Partial Product Matrix (PPM) in an N×N multiplier

Sections of a partial product matrix (PPM) in a N×N multiplier that are exact, hybrid, and approximate are shown in Fig. As illustrated in Fig. 17, the recommended approximation compressors are used to lower the PPM in the approximate part while the traditional full adder and half adder are used to lower the PPM in the exact section. With this method, significant energy savings are made without sacrificing the necessary precision levels. The hybrid portion, which is located between the approximate and exact parts, uses both traditional adder circuits and approximate compressor circuits to lower the PPM. As the multiplier design moves from approximate section to exact section, hybrid columns are needed to balance the carry flow in the PPM. After several reduction phases, the PPM is ultimately reduced to two rows, much like with a traditional multiplier. To lower the PPM in each stage, a mix of exact adders, such as full and half adders, and approximate compressors are used. Additionally, we suggest introducing approximation only in the first and second stages of the PPM reduction sequence in order to reduce the effect of the approximation on the final output. The following are the stages and suggested design technique for putting the N×N approximation multiplier into practice:

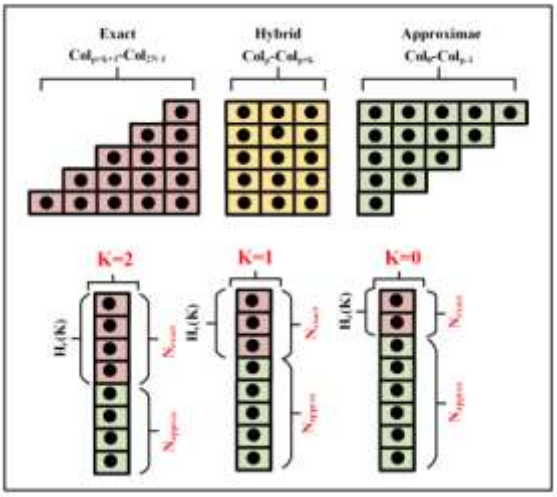

Fig.2 Allocation of compressor & Adder to hybrid columns.

we demonstrate the implementation of the suggested 16 × 16 approximate multiplier (PAxC MUL-16 × 16) and tunable approximate multiplier (PTAxC MUL-16 × 16). As seen in Fig. 19 (a), we start designing the suggested 16 × 16 approximation multiplier (PAxC MUL-16 × 16) and use dot operations between the two input operands to produce the partial product matrix (PPM). According to the calculated PPM, the tallest column's maximum height, or hmax, is
The compressor and adder are assigned to hybrid columns in Figure.

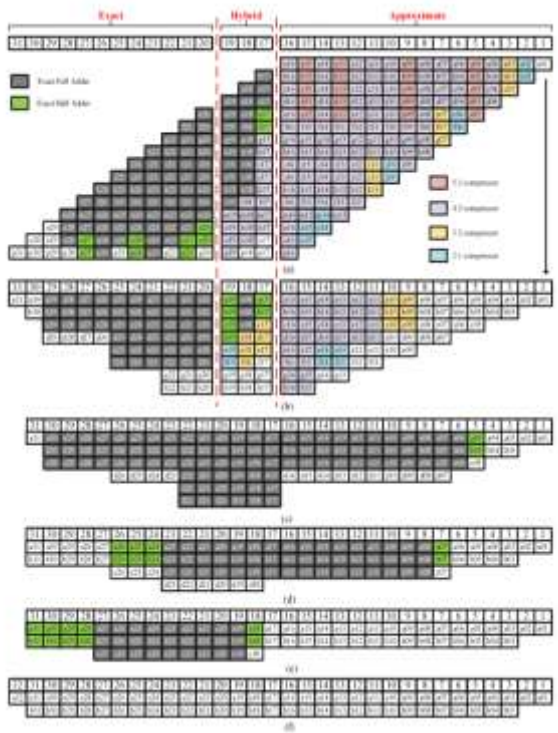

Fig.3.Partial Product reduction stages of the proposed 16 × 16 Multiplier (PAxC MUL-16 × 16)

partial product reduction stages and the distribution of compressor and adder circuits in each step (A) Allocating compressors and adders for Stage 1 reduction (16 → 8) and producing partial products The compressor and adder are assigned for Stage 2 reduction (8 → 6). The allocation of the adder for the reduction of Stage 3 (6 → 4) (d) Stage 4 decrease (4 → 3) adder allocation In Stage 5, the adder is allocated for reduction (3 → 2).(f) Ripple carry adder is used to lower the final two rows of the operand.

1. Determining the Accuracy Needs for Target Applications: Users are instructed to start by determining the target application's computational accuracy needs. Error-tolerant and precision-critical activities should be classified by the user based on the final task. Applications in wearable health monitoring systems, for instance, might accept more errors for less important jobs (like trend analysis) yet require more accuracy for crucial processes (like ECG signal filtering). The suggested adjustable approximate circuits allow designers to experiment with various designs that dynamically modify the approximation levels. This enables the simultaneous optimization of several parameters, including accuracy, delay, and energy usage, in order to achieve certain design objectives.

2. Parameter-Driven Design Space Exploration: Our methodology treats the tunable parameters—such as the working mode of tunable compressors, the degree of logic simplification, and the approximate amount of bits—as tunable knobs. Users can explore the design space and find combinations that meet the necessary computational accuracy by methodically adjusting these parameters.

3. Error-Tolerant Design Flow We incorporate error measurements like Mean Relative Error Distance (MRED) and Normalized Error Distance (NED) straight into the design flow to make sure the designs fulfill the target accuracy standards. During repeated simulations, these measures give designers quantifiable input that helps them fine-tune and choose configurations that accomplish the intended accuracy-energy trade-off.

4. Iterative Design Optimization: Users can test various approximation modes repeatedly and evaluate how they affect applications' accuracy and performance. An ideal solution that strikes a compromise between computational accuracy, power, and performance can be found by designers by comparing the simulation results with accuracy thresholds unique to the application.

5. RESULTS

The suggested approximate compressors (PAxC21, PAxC32, PAxC42, PAxC53), tunable approximate compressors (PTAxC21, PTAxC32, PTAxC42, PTAxC53), approximate multiplier (PAxC MUL 16 × 16), tunable approximate multiplier (PTAxC MUL 16 × 16), approximate MAC unit (PAxC MAC-16), and tunable approximate MAC (PTAxC MAC16) are all thoroughly analyzed in this section.

International Journal of Engineering Research and Science & Technology

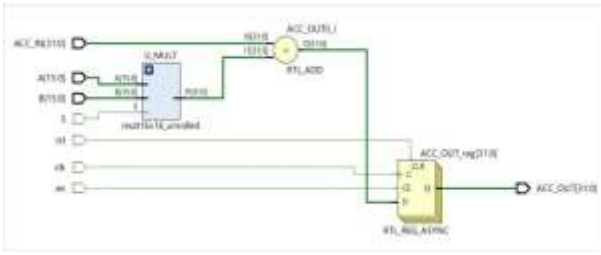

Fig.4.MAC Elaboration Design

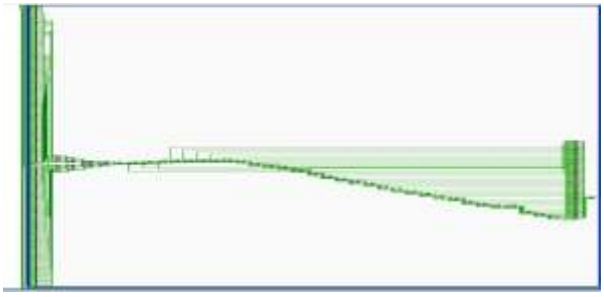

Fig.5.MAC Elaborated design

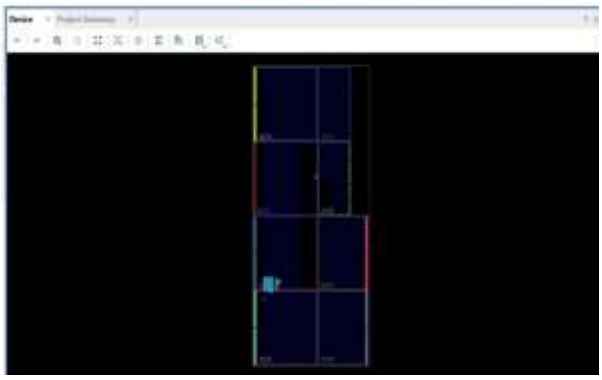

Fig.6.MAC Implementation Design

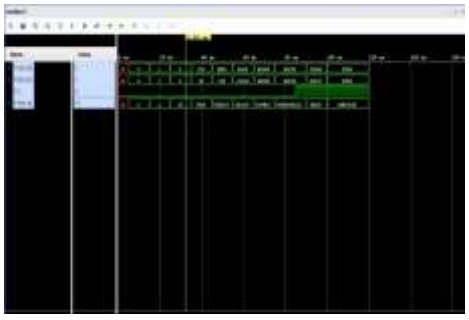

Fig.7.Multiplier Simulation Output

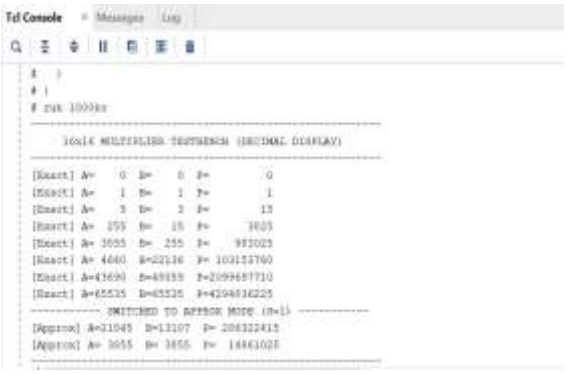

Fig.8.Multiplier Simulation Output

Fig.9.MAC Simulation Output

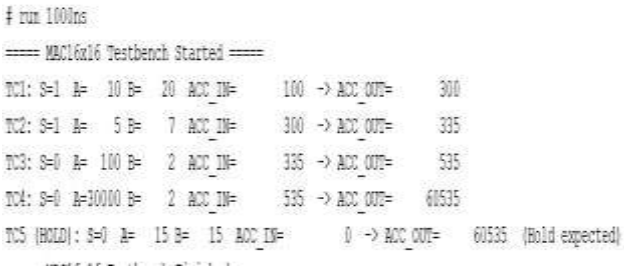

Fig.10.MAC Simulation Output

Utilization	Post-Synthesis	Post-Implementation
		Graph | Table

Resource	Utilization	Available	Utilization %
LUT	374	41000	0.91
FF	32	82000	0.04
IO	97	300	32.33
BUFG	1	32	3.13

Fig.11.Area of MAC

Parameter	Existing Design	Proposed Design
Area (LUTs)	700	374
Power (Watts)	55.508	38.452
Delay (ns)	24.8	9.5

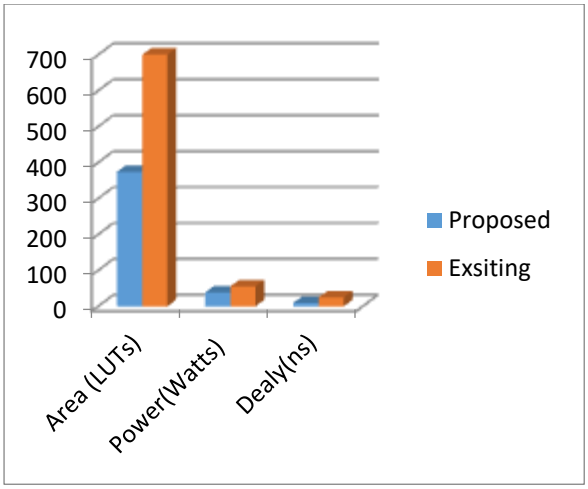

Fig.12.Compression Of Existing And Proposed Mac Units

The suggested architecture's reduced carry propagation and simplified compressor logic result in a 46.6% area reduction. The main causes of the about 30.7% decrease in power usage are decreased logic depth and switching activity. Furthermore, there is a 61.7% decrease in the critical path time, which indicates faster computing and higher throughput. With these enhancements, it is evident that the compressor level configurable approximation greatly improves MAC unit efficiency, which makes the design ideal for Edge AI systems that are energy-constrained and performance-critical.

6. CONCLUSION

For Edge AI applications, this work demonstrated the design and assessment of new adjustable approximation compressor designs for energy-constrained multiply-accumulate (MAC) operations. In order to drastically reduce hardware complexity, power consumption, and critical path time while preserving adequate computational accuracy for error-tolerant workloads, the suggested method makes use of controlled approximation at the compressor level. The suggested design was put into practice and evaluated in terms of area, power consumption, and delay against an existing conventional architecture. Results from experiments show significant gains in all important VLSI measures. The suggested architecture uses just 374 LUTs, which is 46.6% less space than the current design, which takes up 700 LUTs. Reduced carry propagation pathways and streamlined compressor logic are the main causes of this reduction. The suggested design uses 38.452 W of power, which is 30.7% less than the 55.508 W of the current implementation. The reduced power dissipation validates how well approximate computation reduces switching activity and logic depth, which is crucial for edge devices that run on batteries and have limited energy. Additionally, the speed of the suggested architecture is improved by 61.7%, with a much shorter latency of 9.5 ns as opposed to 24.8 ns for the current design. This decreased critical route delay demonstrates how well-suited the suggested compressors are for high-throughput MAC operations needed in real-time Edge AI workloads. Overall, the findings confirm that MAC architectures based on configurable approximation compressors offer a practical way to attain high performance and energy economy, which makes them ideal for next-generation Edge AI accelerators.

References:

[1] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” IEEE European Test Symposium, 2013.

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

[3] O. Akbari et al., “Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers,” IEEE Trans. VLSI Systems, vol. 25, no. 4, pp. 1352–1361, 2017.

[4] R. Zendegani et al., “RoBA multiplier: A rounding-based approximate multiplier,” IEEE Trans. Circuits Syst. I, vol. 64, no. 2, pp. 353–364, 2017.

[5] S. Narayanamoorthy et al., “Energy-efficient approximate multiplication for DSP and machine learning,” IEEE Trans. VLSI Systems, vol. 23, no. 6, pp. 1180–1184, 2015.

[6] T. Rejimon, K. Lingasubramanian, and S. Bhanja, "Probabilistic error modeling for nano-domain logic circuits," IEEE Trans. Nanotechnology, vol. 8, no. 4, pp. 508–516, 2009.
[7] S. Mittal, "A survey of techniques for approximate computing," ACM Computing Surveys, vol. 48, no. 4, 2016.
[8] A. Yazdanbakhsh et al., "ALWANN: Adaptive layer-wise approximation for neural network accelerators," IEEE Trans. VLSI Systems, vol. 27, no. 4, pp. 834–847, 2019.
[9] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Transactions on Computers, vol. 64, no. 4, pp. 984–994, Apr. 2015.
[10] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "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, Apr. 2017.
[11] Naga Charan Nandigama,"A Hybrid Big Data And Cloud-Based Machine Learning Framework For Financial Fraud Detection Using Value-At-Risk", Ijrar - International Journal of Research and Analytical Reviews (IJRAR), E-ISSN 2348-1269, P- ISSN 2349-5138, Volume.11, Issue 3, Page No pp.901-905, September 2024, Available at : http://www.ijrar.org/IJRAR24C3021.pdf.
[12] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "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, Jun. 2015.
[13] N. C. Nandigama, "Data-Warehouse-Enhanced Machine Learning Framework for Multi-Perspective Fraud Detection in Multi-Stakeholder E-Commerce Transactions," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 5, pp. 592–600, May 2023, doi: 10.17762/ijritcc.v11i5.11808.
[14] T. Nanammal and J. Gangrade, "Design of power and area efficient approximate 4:2 compressors for MAC-based signal processing applications," International Journal of Engineering Research & Technology (IJERT), vol. 11, no. 6, pp. 45–50, Jun. 2022.
[15] A. Kumar, R. Patel, and S. Verma, "Comparative analysis of approximate 4:2 compressor designs for low-power multipliers," IEEE International Conference on VLSI Design, pp. 212–217, 2020.
[16] P. Singh and R. Mehra, "Power and area efficient approximate multiplier using compressor-based architecture," Microprocessors and Microsystems, vol. 67, pp. 103–112, 2019.
[17] S. Mittal, "A survey of techniques for approximate computing," ACM Computing Surveys, vol. 48, no. 4, pp. 1–33, Mar. 2016.
[18] S. Shirkavand and M. Moaiyeri, "Accuracy-efficient approximate multipliers using error compensation techniques," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 5, pp. 2315–2319, May 2022.
[19] S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Improving the accuracy of approximate multipliers using error-aware compressor selection," Integration, the VLSI Journal, vol. 91, pp. 102–111, 2023.
[20] J. Kim, S. Lee, and Y. Kim, "Quantization-aware approximate multipliers for energy-efficient edge deep learning accelerators," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 11, pp. 5071–5083, Nov. 2021.