CAAM:Compressor-Based Adaptive Approximate Multiplier with Modified Full Adder (MFA) and Pipelining Technique for Reduced Latency in Neural Network Applications

Dr. Byra Reddy C R[1], Pattapu Vyshnavi[2]

[1]Professor, Department of Electronics and Communication Engineering, Bangalore Institute Of Technology,Bangalore, India

[2.]Student, Department of , Department of Electronics and Communication Engineering, Bangalore Institute Of Technology,Bangalore, India

Abstract*— Approximate computing is a novel approach aimed at enhancing power efficiency, speed, and area utilization in neural networks that can tolerate some error. This letter introduces a new multiplier architecture that employs approximate compressors to minimize errors in partial product columns. By using pipelining and replacing full adders, delay in the Modified Full Adder (MFA) is reduced. An error-correcting module further addresses approximation-induced errors. Experimental results indicate significant improvements in power consumption and power delay product (PDP) for the 8-bit multiplier. The multiplier's effectiveness is confirmed through image processing and neural network applications using the tool as Xilinx Vivado 2022.2, along with a comprehensive analysis of area, power, and timing metrics.

***Keywords*—Approximate compressor, approximate computing, neural network application, partial product reduction (PPR)**

I. INTRODUCTION

This letter proposes a novel unsigned compressor-based adaptive approximate multiplier (CAAM) that utilizes a new methodology for compressor assignment in the partial product reduction (PPR) structure, aiming to minimize circuit complexity while maintaining acceptable accuracy. The goals of the proposed work are as follows:

1. Introduce a new 4:2 approximate compressor circuit to reduce circuit complexity during the PPR stage.
2. Develop an algorithm to identify the most effective approximate compressor from both existing and newly proposed options to reduce errors in the partial product columns.
3. Implement an efficient error-correcting module to mitigate errors introduced by the approximation in the CAAM.

Unlike the proposed CAAM, most existing multipliers, as illustrated in Table 1, do not incorporate an error-correcting module in their designs.

Design	Approximation		No. of region	Schem at LSB region	New Compressor	Error Recovery circuit
	PPG	PPR				
CAAM	No	Yes	3	Compressor Selection	Yes	Yes
Minho	No	Yes	3	No	Yes	No
Zhixi	No	Yes	3	No	Yes	Yes
Xilin	No	Yes	2	No	Yes	No
Multiplier2	No	Yes	2	No	Yes	No
Haroon	Yes	No				No
Mohammad	Yes	No				No
MUL2	No	No	2	No	Yes	Yes
CN	No	Yes	2	No	Yes	No

Table.1: Comparison of different multiplier architectures

A) 4:2 EXACT COMPRESSOR:

A 4:2 exact compressor is a digital circuit used in arithmetic operations, particularly in the multiplication of binary numbers. It is designed to take four input bits and produce two output bits while generating a carry bit that propagates to the next stage in the computation process. The primary function of a 4:2 compressor is to reduce the number of partial products in multiplication, thus simplifying the overall computation and enhancing the speed and efficiency of the operation. As demonstrated in Fig. 1

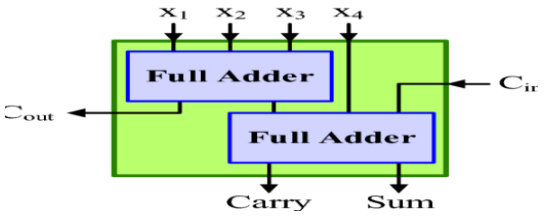


Fig. 1: Schematic diagram for 4:2 Exact Compressor

B) 4:2 APPROXIMATE COMPRESSOR:

A 4:2 approximate compressor is a digital circuit utilized in approximate computing to enhance arithmetic operations, especially in binary multiplications. Unlike exact compressors, it trades off some accuracy to achieve substantial improvements in power consumption, speed, and area. This compressor takes four input bits and produces two output bits and a carry-out bit, using simplified logic to reduce circuit complexity. This reduction leads to fewer transistors, lower power usage, and faster processing times. The minor introduced errors are acceptable. where small inaccuracies do not significantly impact overall performance.

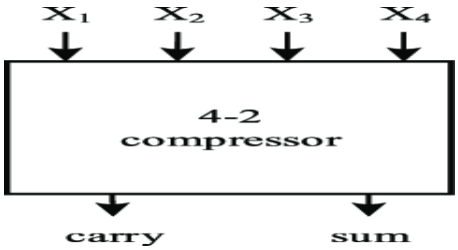


Fig. 2: Schematic diagram for 4:2 Approximate Compressor

The rest of this letter is structured as follows.. Section II covers a range of existing approximate multipliers. Section III presents the proposed design methodology, including the CAAM architecture and the error correction circuit. In Section IV, we provide a comprehensive analysis of error and hardware performance. Finally, conclusions are presented in Section V.

II. LITRATURE SURVEY

[1], In this paper introduces Approximate computing enhances the accuracy-performance tradeoff in error-tolerant applications, with a focus on multiplication. An initial 4:2 compressor with significant error is improved by encoding inputs, reducing faulty truth table rows. Using this, two 4×4 multipliers are designed and scaled to 16×16 and 32×32. The top 16×16 unsigned design achieves a 44% lower power-delay product (PDP), while a radix-4 signed Booth multiplier shows a 52% PDP-MRED reduction. These multipliers excel in image processing and MIMO communication systems, offering high-quality outputs with lower power consumption.
[2], This paper presents Approximate recursive multipliers use lowpower, approximate 4×4 multiplier blocks with varying approximation levels for different sizes. Hybrid partial product-based building blocks consider input operand probability distributions, ensuring efficient hardware and accuracy. High-performance NOR-based half adder (NxHA) and full adder (NxFA) cells are proposed for4×4 multipliers. H. Pei, X. Yi, H. Zhou and Y. He,
[3], The paper introduces approximate computing, which is utilized in digital signal processing applications that can tolerate errors to improve electrical performance. Multipliers, key in computer arithmetic, employ 4-2 compressors to speed up partial product compression. This brief introduces three new approximate 4-2 compressors for 8-bit multipliers and an error-correcting module (ECM) to improve error performance. The compressors, UCAC1, UCAC2, and UCAC3, reduce delay by up to 66.67%, power by up to 93.28%, and area by up to 93.10%, leading to a 49.29% average power reduction in 8-bit multipliers.
[4], Approximate multipliers are extensively studied, with many designs using approximate 4-2 compressors. This paper surveys and compares twelve different approximate 4-2 compressors, including a novel design. These compressors are used to create 8×8 and 16×16 multipliers in 28nm CMOS technology, with configurations for different approximation levels, both signed and unsigned. The study reveals no single best compressor topology, as the optimal choice of depends on required precision, multiplier signedness, and error metrics.
[10], The paper outlines significant progress in Neural Network (NN) that have improved performance for complex tasks, highlighting the growing use of convolutional NNs in embedded systems for image and audio classification, as well as object detection. The multiply-accumulate (MAC) operation is key during NN inference, but integrating thousands of MAC units in NN accelerators significantly boosts power consumption. This work merges approximate computing with NN inference, designing NN-specific approximate multipliers with multiple runtime accuracy levels. A time-efficient framework maps NN weights to these accuracy levels, achieving tight accuracy loss thresholds and substantial energy savings without intensive retraining. Evaluations show 17.8% average energy savings with just a 0.5% loss in inference accuracy.

III. PROPOSED METHODOLOGY

The 8-bit CAAM multiplier architecture, as shown in Fig. 6, is segmented into three distinct regions: 1) a four-bit truncation region, 2) a four-bit approximate region, and 3) a 7-bit accurate region. The key architectural modifications made to these regions are designed to enhance the trade-off between different evaluation metrics, and these improvements will be briefly outlined. The 8-bit CAAM multiplier architecture uses a pipelining technique to process data efficiently. [11]Pipelining enhances instruction throughput by enabling multiple instructions to be executed simultaneously across different stages. This technique organizes the stages of instruction storage and execution in a sequential manner, eliminating feedback loops and thus improving system

concurrency. In this architecture, the pipeline is segmented into multiple stages, with each stage consisting of an input register followed by a combinational circuit. Data flows through these stages, entering at one end and exiting at the other, which leads to increased overall throughput that shows in Fig. 3. The pipelining process reduces the critical path of the system by strategically placing latches, which accelerates operation speeds. This transformation is particularly beneficial for high-speed applications as it minimizes the critical path, although it does result is increased number of latches and system latency. Therefore, integrating pipelining into the CAAM multiplier architecture enhances performance and satisfies the architecture's requirements for efficient data handling and processing. In the CAAM architecture, arithmetic addition is carried out using a modified full adder to improve efficiency as depicted in Fig. 4. Specifically, this modified full adder employs two 4:1 multiplexers to streamline its design. Each multiplexer is governed by selection lines 'a' and 'b'. The input 'C' to one multiplexer is used to generate the sum output, while the other multiplexer, with inputs 0, 'C', and 1, produces the carry output. This modification contrasts with a conventional full adder by significantly reducing the gate count, thereby resulting in a more efficient implementation .Furthermore, the use of multiplexers in this context leverages their combinational circuit capabilities. A multiplexer (Mux) can manage up to 2^n data inputs, where 'n' represents the number of selection lines, and has a single output line. The output is determined by the values of the selection lines, which choose one of the data inputs. For instance, a 4x1 multiplexer has four data inputs (I_3, I_2, I_1, I_0), two selection lines (s_1, s_0), and one output (Y), as illustrated in Fig. 5. This architecture allows for a more streamlined and efficient approach to arithmetic operations within the CAAM system.

A) Pipelined structure:

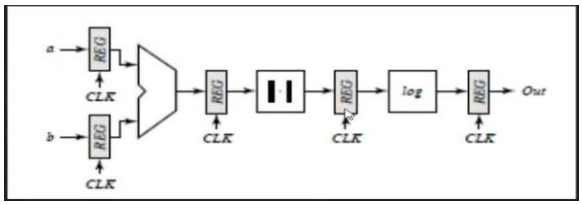


Fig 3: Pipelined Design

B) MFA (Modified Full Adder):

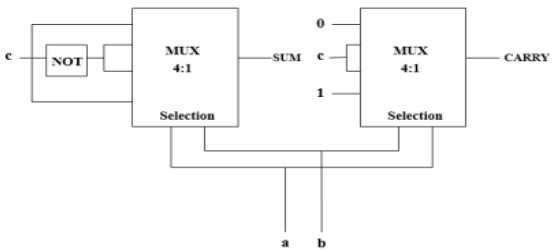


Fig. 4: Modified Full Adder

C) MUX:

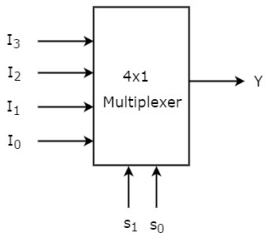


Fig. 5: 4:1 multiplexer

D) PPR(Partial Product Reduction):

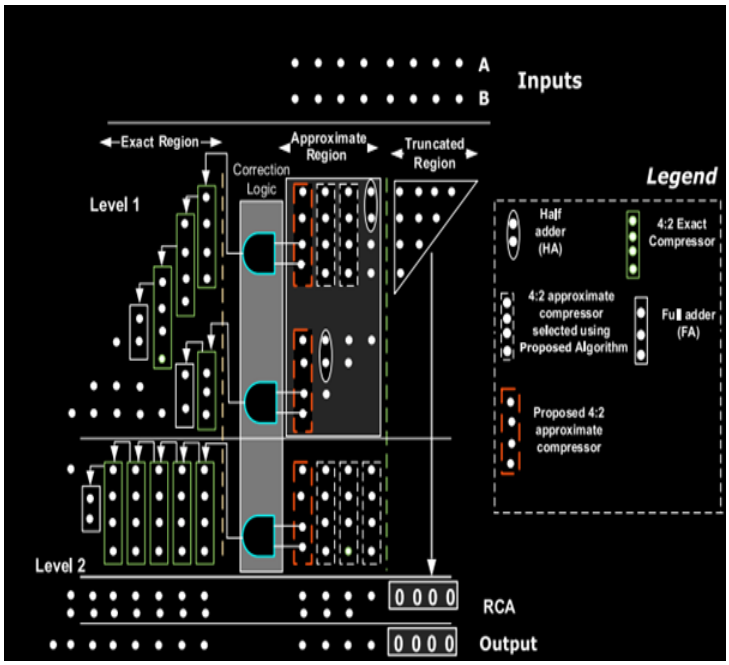
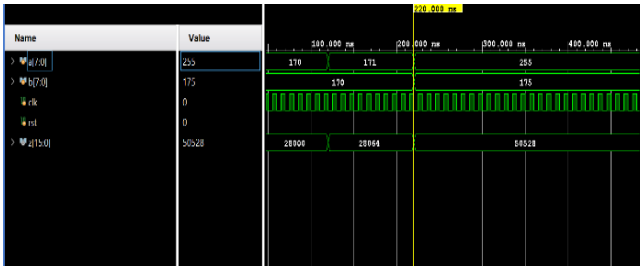


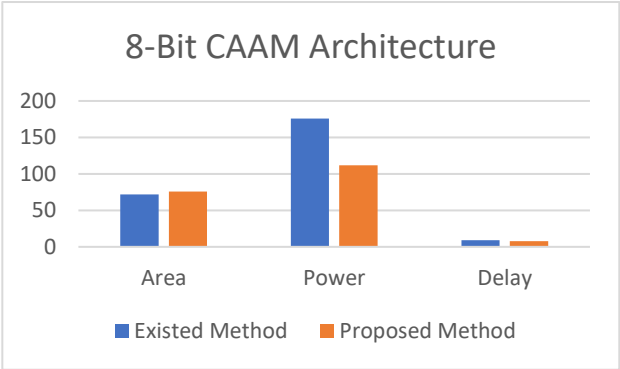
Fig. 6: PPR structure of CAAM design

IV. Experimental Result And Discussion

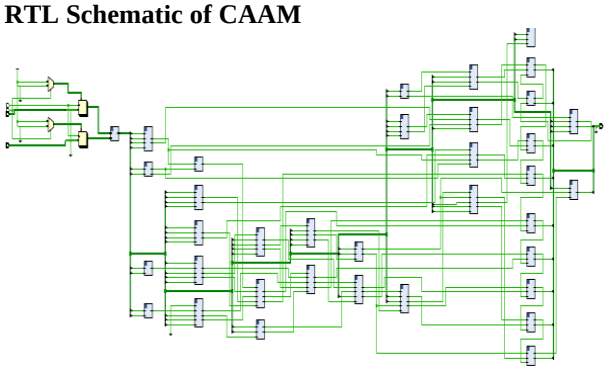
Approximate computing takes advantage of the error tolerant nature of neural network applications to decrease computational complexity. This letter introduces an unsigned multiplier designed to streamline computation by pruning the least significant region(LSR)and approximating the middle region within the PPR structure. Additionally, a simple algorithm is introduced to effectively compress the partial product columns (PPC) within the approximate region. Detailed hardware results demonstrate that the CAAM architecture achieves a 36.6% reduction in power consumption and a 14.44% improvement in operational performance compared to existing designs.[11]. Validation confirms that CAAM offers a more efficient balance between computational quality and effort in neural networkapplications.

A) Simulation waveform of CAAM

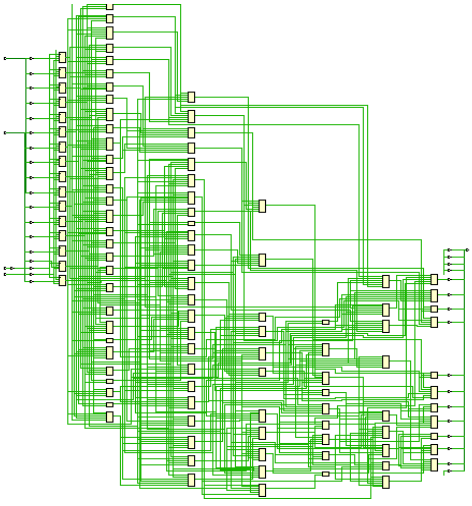




B) RTL Schematic of CAAM



C) Technological Schematic of CAAM



D) Synthesis Repot for CAAM Architecture.

8-Bit CAAM Architecture	AREA	POWER	DELAY
Existing Method	72	176mW	9.004nS
Proposed Method	76	112mW	7.704nS

E) Comparsion between Exsisting method and Proposed Method

REFERENCES

[1] M. S. Ansari, H. Jiang, B. F. Cockburn, and J. Han, "Low-power approximate multipliers using encoded partial products and approximate compressors," IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 8, no. 3, pp. 404–416, Sep. 2018, doi: 10.1109/JETCAS.2018.2832204.

[2] H. Waris, C. Wang, W. Liu, J. Han, and F. Lombardi, "Hybrid partial product-based high-performance approximate recursive multipliers," IEEE Trans. Emerg. Topics Comput., vol. 10, no. 1, pp. 507–513, Jan.–Mar. 2022, doi: 10.1109/TETC.2020.3013977.

[3] H. Pei, X. Yi, H. Zhou, and Y. He, "Design of ultra-low power consumption approximate 4–2 compressors based on the compensation characteristic," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 68, no. 1, pp. 461–465, Jan. 2021, doi: 10.1109/TCSII.2020.3004929.

[4] 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, doi: 10.1109/TCSI.2020.2988353.

[5] H. Waris, C. Wang, C. Xu, and W. Liu, "AxRMs: Approximate recursive multipliers using high-performance building blocks," IEEE Trans. Emerg. Topics Comput., vol. 10, no. 2, pp. 1229–1235, Apr.–Jun. 2022, doi: 10.1109/TETC.2021.3096515.

[6] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for errorresilient multiplier design," in Proc. IEEE Int. Symp. DFTS, 2015, pp. 183–186, doi: 10.1109/DFT.2015.7315159.

[7] M. Ha and S. Lee, "Multipliers with approximate 4–2 compressors and error recovery modules," IEEE Embedded Syst. Lett., vol. 10, no. 1, pp. 6–9, Mar. 2018, doi: 10.1109/LES.2017.2746084.

[8] S. Venkatachalam and S. Ko, "Design of power and area efficient approximate multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 5, pp. 1782–1786, May 2017, doi: 10.1109/TVLSI.2016.2643639.

[9] X. Yi, H. Pei, Z. Zhang, H. Zhou, and Y. He, "Design of an energyefficient approximate compressor for error-resilient multiplications," in Proc. IEEE ISCAS, 2019, pp. 1–5, doi: 10.1109/ISCAS.2019.8702199.

[10] Z. -G. Tasoulas, G. Zervakis, I. Anagnostopoulos, H. Amrouch, and J. Henkel, "Weight-oriented approximation for energy-efficient neural network inference accelerators," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 12, pp. 4670–4683, Dec. 2020, doi: 10.1109/TCSI.2020.3019460.

[11] S Vignesh Bharadwaj, Avinash Bhat Pattaje, Suresh Nambi , and Syed Ershad Ahmed, "CAAM: Compressor-Based Adaptive Approximate Multiplier for Neural Network Applications," IEEE EMBEDDED SYSTEMS LETTERS, VOL. 15, NO. 3, SEPTEMBER 2023