SELF-ERROR CORRECTION ACCUMULATION MODULES IN MULTIPLY-ACCUMULATE UNIT DESIGN FOR ENHANCED ACCURACY

KANCHI SAI KARTHIK[1], SHILPA K[2]

[1]PG Scholar, [2]Assistant Professor, Branch: VLSI-SD

J.B. INSTITUTE OF TECHNOLOGY AND MANAGEMENT, Hyderabad, Telangana

Abstract: In the realm of digital signal processing, the design of Multiply-Accumulate (MAC) units plays a pivotal role in achieving high-performance computations. The MAC design finds application in various fields such as digital signal processing, communications, and artificial intelligence, where MAC operations are fundamental for efficient computation of convolutional neural networks, digital filters, and other signal processing tasks. Currently, MAC units commonly rely on separate adders and multipliers, leading to increased hardware complexity and power consumption. The existing systems often struggle to strike a balance between speed and resource utilization. Traditional MAC designs entail redundant hardware due to the independent implementation of adders and multipliers. Separate adders and multipliers contribute to higher power consumption, limiting energy efficiency. The existing systems may face scalability challenges, especially when aiming for high-performance computing, due to their architecture. This work introduces a novel approach to MAC unit design by employing unified adders and multipliers, aiming to enhance both speed and resource utilization. The proposed method integrates unified adders and multipliers, optimizing the MAC unit for improved speed and efficiency. By leveraging a unified architecture, the design minimizes redundancy, enhances resource utilization, and reduces power consumption

Keywords: MAC unit, redundancy, high-performance computing, enhances resource utilization, unified architecture.

I. INTRODUCTION

In VLSI circuits, the MAC is a crucial component, especially in Digital Signal Processing (DSP) and numerical computations. The MAC is dual operand digital signal processing instructions. MAC is considered important in all DSP architectures. It comprises of a multiplier, adder, and accumulator, efficiently performing multiplication and addition operations in various mathematical algorithms.

Over the years, researchers and engineers have actively proposed various ideas to enhance MAC unit performance. One significant focus has been on addressing the challenges posed by excessive partial product term generation during conventional multiplication approaches. The innovation in MAC unit design includes the incorporation of self-error correction and accumulation modules. These modules contribute to real-time error detection and rectification, improving precision and reducing accumulation errors. This MAC unit is valuable in applications such as image recognition systems, medical imaging, and audio processing.

The applications of the innovative MAC unit with self-error correction and accumulation modules extend beyond communications to image and signal processing. Real-time error correction enhances its value in image recognition, medical imaging, and audio processing applications. The improved precision significantly refines the quality of processed images and signals, impacting industries from healthcare to multimedia. In addition to its pivotal role in communications, image

recognition, medical imaging, and audio processing, this MAC unit introduces a paradigm shift in computational capabilities. Its adaptability and precision make it a promising candidate for integration into emerging fields such as artificial intelligence (AI). In AI applications, where real-time processing and error resilience are critical, the MAC unit's features align seamlessly with the demands of complex algorithms and data-intensive tasks. This versatility positions the MAC unit as a cornerstone in the development of AI systems, opening new possibilities for improved accuracy and efficiency in AI-driven processes.

The MAC's significance lies in its ability to handle complex calculations with improved speed and accuracy. Speed, area and performance are the major constraints that have to be considered. It is a key element in processors and DSPs within VLSI design, optimized for multiply-accumulate operations prevalent in applications such as audio processing, image processing, and communications. The design of the MAC unit is guided by the need for a balance between speed and area optimization. The speed of the multiplier determines the critical path, and efficient area utilization is essential for effective MAC unit design. Moreover, performance metrics directly impact the overall efficiency and speed of the VLSI system. Precision and speed are carefully considered to meet the specific requirements of targeted applications. Power consumption is a critical consideration, and this incorporates an efficient algorithms and hardware optimizations to minimize power usage while maintaining desired computational capabilities.

II. LITERATURE SURVEY

The literature survey demonstrates viewpoints, methodological solutions and research results related to the area. The existing information is critically analyzed so that contradicting and differing research methods are shown. Only material that is relevant and directly related to the research is selected in the survey. Designing an efficient MAC unit involves addressing trade-offs between performance, area, and power consumption, which are essential considerations in VLSI design. MAC units are often implemented using multiplier and adder circuits. Various architectures exist, including serial and parallel implementations, depending on the specific requirements of the application and the desired trade-offs between speed and area.

Di Meo, et.al [1] investigated a MAC unit which computed Y = A×B+C using static segmentation. The proposed architecture used a unique carry-propagate adder and performed segmentation on the three operands A, B, and C, to reduce hardware cost. The circuit could be configured at design-time by two parameters. The first one controlled the segmentation on A and B, while the second one controlled the segmentation on C and the adder length. An error compensation technique was also employed to reduce the approximation error. Error analysis and implementation results in 28nm CMOS for 8-bits multiplier with 20-bits and 24-bits addition were presented. The proposed approximate MACs outperformed the state of the art, showing the largest power saving when the mean relative error distance (MRED) was larger than 2×10^{-3} and 4×10^{-5} for 20 and 24-bits addition, respectively. For MRED of about 6×10^{-3}, the proposed approximate MAC with 20-bits addition exhibited a power reduction larger than 60% compared to the exact MAC and larger than 27% compared to the state-of-the-art approximate MACs. Application examples to image filtering and template matching showed that proposed approximate circuits were good candidates in applications where their error performances were acceptable.

Zhang, et.al [2] proposed a Hybrid CAM-MAC RRAM-based Accelerator (HyAcc) to address the challenges of the embedding layer. Firstly, they recognized that content-addressable-memory (CAM) crossbar could broadcast the input item IDs across all

rows to gather the stored item IDs at one cycle. Hence, they designed RRAM-based CAM crossbars to gather item IDs efficiently. In the meantime, they utilized the multiplication-and-accumulation (MAC) crossbars to implement the reduction operation in the embedding layer. Whereas, during the gather operation, the RRAM-based CAM crossbar inevitably encountered the access inefficiency problem because only one item ID could be gathered per cycle. To overcome this, they proposed the hot/cold item engines containing fine-grained/coarse-grained CAM crossbars for the input item IDs with high-frequency/low-frequency (termed as hot/cold item IDs). Additionally, since the input cold item IDs were unevenly distributed in the coarse-grained CAM crossbars, they may have caused the workload imbalance problem. To alleviate it, they presented the access-aware dynamic pruning solution to dynamically prune the redundant input cold item IDs and average the workload of the coarse-grained CAM crossbars. Extensive experiments validated the effectiveness of the proposed HyAcc architecture.

Kim, et.al [3] presented a design that improved tolerance against process variation with a smaller cell area compared to previous capacitive SRAM CIM designs while inheriting the advantage of capacitive SRAM CIM hardware such as the linearity in MAC results and suppression of the static readout current. They also demonstrated a compact and low-power ADC for CIM readout, which improved the energy efficiency significantly. Finally, they demonstrated a programmable on-chip ADC reference voltage generator circuit for adjusting the ADC input range using bitcell replica arrays. The proposed circuit reduced the ADC bit-resolution requirement by considering the distribution of MAC results and also helped to address the effect of the parasitic bitline capacitance. Measurement results showed that a 128×128 macro fabricated in a 28 nm CMOS achieved 1519.5 TOPS/W at 0.7 V.

Subin ki, et.al [4] introduced an accelerator that employed a hardware-friendly shift-based floating-fixed MAC operator and shift-based quantization method that significantly reduced hardware resources and minimized accuracy degradation. The pipelined streamline architecture maximized hardware utilization and stored all parameters in on-chip memory to minimize external memory access. Moreover, the Gaussian modeling-based performance enhancement technique was effectively processed in the programmable system to address the low accuracy issue in lightweight models. The proposed IP, implemented on Xilinx XCVU9P, achieved a processing speed of 62.9 FPS and an accuracy of 34.01% on the COCO2014 dataset, which demonstrated the superiority of the proposed accelerator over prior research in terms of the trade-off between throughput, hardware resources, and model accuracy.

III. RESEARCH MOTIVATION

Figure 1 shows the research motivation. In VLSI design, the motivation and process involve creating complex integrated circuits (ICs) that can contain millions or even billions of transistors on a single chip. The primary goals of VLSI design are to enhance functionality, performance, and energy efficiency while minimizing size and cost.

A microprocessor is an integrated circuit that contains the logic and control circuitry required to perform the functions of a computer's central processing unit It is a multipurpose, clock-driven, register-based, digital integrated circuit that accepts binary data as input, processes it according to instructions stored in its memory, and provides results as output.

A MAC operation is a common operation in DSP and other computational tasks. It involves multiplying two numbers and accumulating the result. The MAC operation is often used in applications such as correlation, convolution, and matrix operations.

The above figure shows the success or failure of a microprocessor depends on the

major constraints such as power consumption, area, and delay of the MAC unit. If the MAC unit consumes more power, area, and delay, then the microprocessor fails. However, if the MAC unit is efficient in terms of power consumption, area, and delay, then the microprocessor succeeds.

So, the motivation in VLSI design revolves around achieving higher integration density, improving performance, reducing power consumption, lowering costs, ensuring reliability, and leveraging automation to cope with the increasing complexity of modern semiconductor devices. The VLSI design process involves a multidisciplinary approach, combining engineering principles, computer-aided design tools, and advanced manufacturing technologies to create sophisticated ICs.

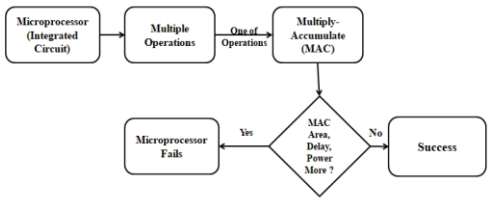

Fig1: Research Motivation for MAC Design.

IV. EXISTING SYSTEM

In the existing system of MAC, a digital combinational circuit is utilized for multiplying two binary numbers by employing an array of full adders and half adders. This array facilitates the nearly simultaneous addition of various product terms. To generate these product terms, an array of AND gates is employed before the Adder array. The fundamental operation of an array multiplier involves generating partial products by multiplying the multiplicand with each multiplier bit. These partial products are then shifted according to their bit orders and subsequently added together. The performance of an array multiplier can be analyzed based on parameters such as silicon area, delay, and power consumption. Researchers have concentrated on design factors encompassing high speed, accuracy, low power consumption, regularity of layout, and minimized area usage. The ongoing efforts in research aim to enhance the efficiency of array multipliers by addressing these factors, contributing to the continual evolution of digital combinational circuits for multiplication operations.

In addition, the existing system of MAC design, ripple carry adders are commonly employed. These adders serve as a key component in the construction of the multiplier module within the MAC unit. It is essential to acknowledge that the incorporation of ripple carry adders in MAC design can vary based on the particular implementation and optimization objectives. Diverse designs and techniques may be utilized to enhance performance metrics such as power consumption, speed, and area utilization. Researchers and designers continually explore innovative approaches to achieve superior efficiency in MAC units, tailoring the use of ripple carry adders to meet specific performance goals and system requirements.

Moreover, the existing system of MAC design, basic multipliers and adders are frequently utilized. A standard MAC unit typically incorporates an adder, a multiplier, and an accumulator. The adders employed in MAC units are commonly either Carry-Select or Carry-Save adders, emphasizing speed, which is crucial in DSP applications. The choice of multipliers in MAC units can vary, contingent upon the particular implementation and optimization objectives. Researchers and designers explore diverse multiplier designs to enhance efficiency, tailoring their selection to specific performance goals and system requirements.

V. PROBLEM STATEMENT

The design of ICs for MAC units in VLSI poses significant challenges in terms of achieving high performance, low power consumption, and efficient area utilization. MAC units are fundamental building blocks in various signal processing applications, such as DSP, image processing, and communication systems. Existing MAC system in VLSI causes higher area and Power consumption followed by higher time delay.

One major challenge in MAC unit design is the need for high-speed multiplication and accumulation operations. MAC system involves higher complexity in design. Achieving high performance without compromising accuracy is crucial, especially in real-time processing applications is very difficult due to its system design.

Furthermore, the integration of MAC units into larger VLSI systems requires careful consideration of interconnectivity and data flow. Normal MAC system involves improper data analysis, bus architectures, and interconnect schemes are critical to minimizing delays and optimizing overall system performance. Ensuring that the MAC unit interfaces seamlessly with other components in the system is essential for achieving reliable and predictable operation.

Key challenge in MAC unit design for VLSI is the demand for low power consumption. As portable electronic devices and battery-operated systems become increasingly prevalent, power efficiency is a critical factor in the success of ICs. Designers must employ multiplication and accumulation operations. Achieving high accuracy without sacrificing speed is that requires careful consideration of the underlying mathematical models and the precision of the implemented hardware.

VI. RESEARCH OBJECTIVE

The existing MAC units consume significant area, experience delays, and consume high power, which can lead to failures of microprocessors. This project aims to overcome these challenges by introducing self-error correction and accumulation modules within the MAC unit itself, without the need for external support. advancement is envisioned to prevent system failures and elevate computation efficiency across various applications. Additionally, this project focuses on optimizing the area, delay and power consumption of the MAC unit to enhance the overall efficiency and performance. It aims to minimize the delay introduced by mac units by optimizing the critical path and employing self-error correction and accumulation modules.

VI. RESULTS

Below figure shows the existing simulation results for N=32 bit. Here, ‘a’ and ‘b’ are the inputs, ‘en’ is the enable signal and should always be in a active state and ‘clk' represents clock cycle. The Multiplier first performs the multiplication operation on these inputs. Then result is then sent to an adder, which adds this product to the accumulated sum. This accumulated sum is stored in an accumulator. After the addition, the accumulated sum is fed back to the adder, where it is added to the next product of ‘a’ and ‘b’. This process continues and final result is stored in the accumulator representing the sum of all the products of ‘a’ and ‘b’.

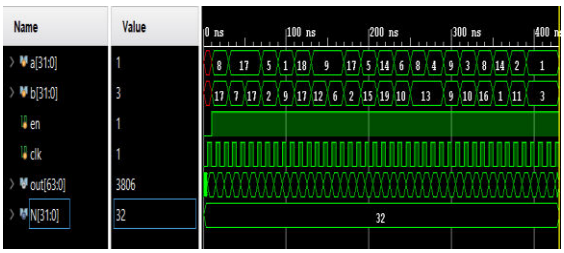

Fig 2 Existing Simulation Results for N=32

Fig 3 shows the existing power measurements for N=32. Here, the total power is 103.437 µw, Static power includes PL Static power of 1.235 µw, Dynamic power includes Signals power of 17.083 µw, Logic power of 25.080 µw and I/O power of 60.039 µw.

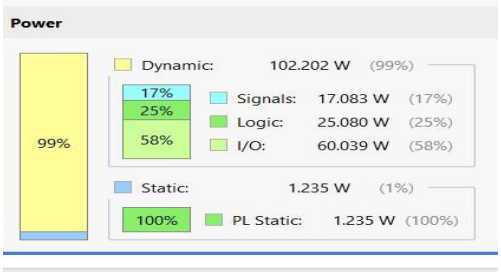

Fig 3: Existing Power for N=32

Proposed Result

Fig 4 shows the existing simulation results for N=32 bit. Here, ‘a’ and ‘b’ are the inputs, ‘en’ is the enable signal and should always be in a active state and ‘clk' represents clock cycle. The Multiplier first performs the multiplication operation on these inputs. Then result is then sent to an adder, which adds this

product to the accumulated sum. This accumulated sum is stored in an accumulator. After the addition, the accumulated sum is fed back to the adder, where it is added to the next product of 'a' and 'b'. This process continues and final result is stored in the accumulator representing the sum of all the products of 'a' and 'b'.

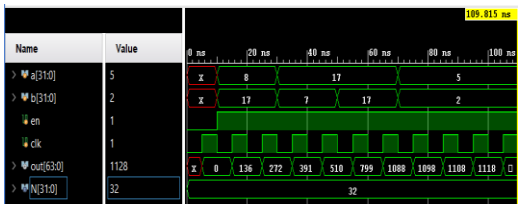

Fig 4: Proposed Simulation Results for N=32

Table shows the proposed area measurements for N=32. Here, 1323 number of LUT's are used out of Available 133800 LUT's which consumes 0.99% of utilization, 64 number of FF's are used out of Available 267600 FF's which consumes 0.02% of utilization, 130 number of IO's are used out of Available 500 IO's which consumes 26.00% of utilization, 1 number of BUFG's are used out of Available 32 BUFG's which consumes 3.13% of utilization.

Resource	Utilization	Available	Utilization %
LUT	1323	133800	0.99
FF	64	267600	0.02
IO	130	500	26.00
BUFG	1	32	3.13

Table Proposed Area for N=32

Power

Fig 5 shows the proposed power measurements for N=32. Here, the total power is 81.64 µw, Static power includes PL Static power of 1.235 µw, Dynamic power includes Signals power of 10.218 µw, Logic power of 10.090µw and I/O power of 60.098 µw.

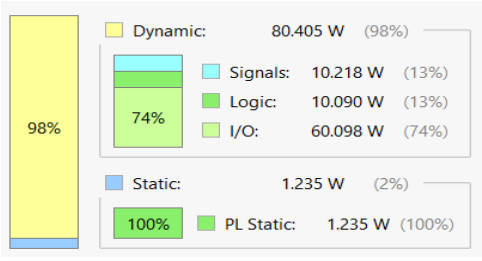

Fig 5: Proposed Power for N=32

CONCLUSION

The proposed Multiply-Accumulate (MAC) architecture presents a highly efficient and reliable solution for arithmetic operations, particularly in digital signal processing and machine learning. By leveraging a Radix-4 Modified Booth Multiplier (MBM), the architecture minimizes partial product rows, reducing hardware complexity and improving overall efficiency. This, combined with an Error Correctable Carry Look Ahead Adder (EC-CLA) for accumulation, ensures fast and accurate computation of the final result. Additionally, the architecture's inclusion of data storage units enables pipelining and parallel processing, further enhancing performance and throughput. Moreover, the MAC architecture's versatility is highlighted by its support for both positive and negative numbers, as well as its scalability to different radices. The incorporation of error correction mechanisms in the EC-CLA ensures data integrity, adding a layer of reliability to the computation process. These features make the architecture suitable for a wide array of high-performance computing applications, where complex arithmetic operations need to be executed with minimal latency and maximum efficiency.

In conclusion, the proposed MAC architecture stands out as a robust and adaptable solution for arithmetic operations in various domains. Its efficient use of the Radix-4 Modified Booth Multiplier and the Error Correctable Carry Look Ahead Adder, coupled with its support for different radices and error correction mechanisms, makes it a compelling choice for high-performance computing tasks. Overall, the architecture's ability to deliver fast, accurate, and reliable computation makes it a valuable addition to the field of digital signal processing and machine learning.

FUTURE SCOPE

The future scope of the proposed MAC architecture is vast and holds tremendous potential for revolutionizing various fields. Its efficient design and versatile capabilities make it a compelling solution for a

wide range of applications, promising advancements in machine learning, digital signal processing, robotics, IoT, and beyond. The future of the MAC architecture is bright, promising innovations that can drive efficiency, performance, and reliability in a wide range of applications, ultimately shaping the future of computing.

Increased Integration with Machine Learning Systems: - The MAC architecture's efficiency and reliability make it well-suited for integration into machine learning systems, particularly in areas such as neural network training and inference, where complex arithmetic operations are frequently performed.

Enhanced Energy Efficiency: - Future developments could focus on further improving the energy efficiency of the MAC architecture, making it more sustainable and suitable for use in energy-constrained environments such as IoT devices and edge computing systems.

Exploration of Alternative Multiplier and Adder Architectures: - Research could explore alternative multiplier and adder architectures to further enhance the performance and efficiency of the MAC design, potentially leading to even more optimized solutions for arithmetic operations.

References

[1]. Di Meo, Gennaro, Gerardo Saggese, Antonio GM Strollo, and Davide De Caro. "Approximate MAC unit using Static Segmentation." *IEEE Transactions on Emerging Topics in Computing* (2023).

[2]. Zhang, Xuan, Zhuoran Song, Xing Li, Zhezhi He, Li Jiang, Naifeng Jing, and Xiaoyao Liang. "HyAcc: A Hybrid CAM-MAC RRAM-based Accelerator for Recommendation Model." In *2023 IEEE 41st International Conference on Computer Design (ICCD)*, pp. 375-382. IEEE, 2023.

[3]. Kim, Eunhwan, Hyunmyung Oh, Nameun Kang, Jihoon Park, and Jae-Joon Kim. "A Capacitive Computing-In-Memory Circuit with Low Input Loading SRAM Bitcell and Adjustable ADC Input Range." *IEEE Transactions on Circuits and Systems II: Express Briefs* (2023).

[4]. Subin Ki, Juntae Park, and Hyun Kim. "Dedicated FPGA Implementation of the Gaussian TinyYOLOv3 Accelerator." *IEEE Transactions on Circuits and Systems II: Express Briefs* (2023).

[5]. Yao, Chun-Yen, Tsung-Yen Wu, Han-Chung Liang, Yu-Kai Chen, and Tsung-Te Liu. "A Fully Bit-Flexible Computation in Memory Macro Using Multi-Functional Computing Bit Cell and Embedded Input Sparsity Sensing." *IEEE Journal of Solid-State Circuits* (2023).

[6]. Shubham Kumar, Paul R. Genssler, Somaya Mansour, Yogesh Singh Chauhan, and Hussam Amrouch. "Frontiers in AI Acceleration: From Approximate Computing to FeFET Monolithic 3D Integration." In *2023 IFIP/IEEE 31st International Conference on Very Large-Scale Integration (VLSI-SoC)*, pp. 1-6. IEEE, 2023.

[7]. Cheon, Sungsoo, Kyeongho Lee, and Jongsun Park. "A 2941-TOPS/W Charge-Domain 10T SRAM Compute-in-Memory for Ternary Neural Network." *IEEE Transactions on Circuits and Systems I: Regular Papers* (2023).

[8]. Wang, Shuyu, and Hao Cai. "Computing-in-Memory with Enhanced STT-MRAM Readout Margin." *IEEE Transactions on Magnetics* (2023).

[9]. Jing, Naifeng, Zihan Zhang, Yongshuai Sun, Pengyu Liu, Liyan Chen, Qin Wang, and Jianfei Jiang. "Exploiting bit sparsity in both activation and weight in neural networks accelerators." *Integration* 88 (2023): 400-409.

[10]. Antolini, Alessio, Carmine Paolino, Francesco Zavalloni, Andrea Lico, Eleonora Franchi Scarselli, Mauro Mangia, Fabio Pareschi et al. "Combined HW/SW Drift and Variability Mitigation for PCM-based Analog In-memory Computing for Neural Network Applications." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 13, no. 1 (2023): 395-407.

Author

KANCHI SAI KARTHIKcompleted Bachelor's degree and studying M. Tech in the branch of VLSI-SD from J.B. INSTITUTE OF TECHNOLOGY AND MANAGEMENT, Hyderabad, Telangana.