International Journal of Engineering Research and Science & Technology

IJERST

ISSN : 2319-5991

www.ijerst.com

Email: editor@ijerst.com or editor.ijerst@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Design of Multiply Accumulate Unit with Self-ErrorCorrection and Accumulation Modules

G. Sri Lakshmi [1], S. Sai Santosh [2], P. Sai Kiran [3], Dr. N. Sowmya [4]
[1,2,3] UG Scholar, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
[4]Assistant Professor, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
gadisrilakshmi.004@gmail.com

Abstract:

Multiply-Accumulate Units (MACs) are essential components in digital signal processing (DSP) and machine learning algorithms, widely employed in convolutional neural networks (CNNs) and other computations. Current research shows that enhancing the efficiency of MAC units can lead to a 20-30% reduction in computation time, while power consumption can decrease by 15-25%, directly benefiting various high-performance systems. Traditional MAC units rely on simple adders and multipliers, which suffer from limited precision, slower speed, and higher power consumption in large-scale implementations. This work presents a novel MAC design incorporating a Modified Booth Multiplier and an Error Correctable Carry Look Ahead Adder (ECCLA). The Modified Booth Multiplier enhances the multiplication speed by reducing partial products, while the ECCLA improves precision and fault tolerance during accumulation. The proposed MAC architecture also features self-error correction mechanisms that detect and correct arithmetic errors, ensuring improved performance and reliability in error-sensitive applications. Experimental results demonstrate that the proposed MAC design achieves up to 35% faster computation speed and 20% more accurate results compared to traditional architectures.

Keywords: Multiply and Accumulate Unit, Digital Signal Processing, Convolutional Neural Networks, Modified Booth Multiplier, Error Correctable Carry Look Ahead Adder, Self Error Correction.

1. 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. 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. 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.

2. LITERATURE SURVEY

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 was 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 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 was 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 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

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

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 the process variation with a smaller area compared to previous SRAMM 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.

Yao, et.al [5] presented a CIM macro that employed a literature multi-functional computing bit cell design by integrating the MAC and the A/D conversion to maximize efficiency and flexibility. Moreover, an embedded input sparsity sensing and a self-adaptive dynamic range (DR) scheme were proposed to minimize the energy- consuming A/D conversions in CIM. Finally, the CIM macro implementation utilized an interleaved placement structure to enhance the weight-updating bandwidth and the layout symmetry. The CIM design, fabricated in standard 28-nm CMOS technology, achieved an area efficiency of 27.7 TOPS/mm2 and an energy efficiency of 291 TOPS/W, demonstrating a highly energy-area- efficient flexible CIM solution.

Shubham Kumar, et.al [6] proposed a work where they explored the performance and energy advantages of employing classical AI acceleration with conventional systolic MAC arrays. They highlighted the growing importance of monolithic 3D integration as a transformative hardware acceleration strategy, moving beyond the constraints of classical von Neumann architectures. They also discussed how brain-inspired hyperdimensional computing (HDC) offered an exciting avenue for overcoming the power-hungry requirements often associated with MAC arrays, which were inevitable in deep learning hardware. Addressing the limitations of von Neumann architectures, they presented the potential of monolithic 3D integration to enable ultra-dense Processing-in- Memory (PiM) layers stacked on top of high-performance CMOS logic. This literature approach offered to enhance computational performance. Recognizing the need for compatibility with low thermal budgets, they identified ferroelectric thin-film transistors (FeTFT) as a promising candidate for back-end-of-line (BEOL) fabrication. They highlighted recent advances in BEOL FeTFT technology and demonstrated how technology/algorithm co- optimization played a crucial role in the successful realization of reliable brain-inspired HDC on potentially unreliable FeTFT-based PiM layers. Their results showcased the potential of these innovations for the development of next-generation, energy-efficient AI hardware.

Cheon, et.al [7] proposed a 10T SRAM bitcell, employing charge- domain analog computations to improve the noise tolerance of bit- line (BL) signals, where the MAC results were represented in CiM. Parallel processing of three different analog levels for ternary input activations was also performed in the proposed single 10T bitcell. To reduce the analog-to-digital converter (ADC) bit-resolutions without sacrificing deep neural network (DNN) accuracies, a confined-slope non-uniform integration (CS-NUI) ADC was proposed, which provide layer-wise adaptive quantization for multiple different layers with different MAC distributions. Additionally, by sharing the ADC reference voltage generator in every single column of the SRAM array, the ADC area was effectively reduced with improved energy efficiencies of CiM. The 256×64.10T SRAM CiM macro with the proposed charge-sharing scheme and CS-NUI ADCs was implemented using 28nm CMOS process. The silicon measurement results showed that the proposed CiM exhibited accuracies of 98.66% and 88.48% with MNIST dataset on MLP, and CIFAR-10 dataset on VGGNet-7.

3. PROPOSED METHODOLOGY

In the world of VLSI circuits, the integration of a MAC utilizing a Booth multiplier and a Carry Look-Ahead Adder (CLA) represents a significant advancement in digital signal processing and numerical computations. The Booth multiplier is a specialized algorithmic approach that optimizes the multiplication process by reducing the number of partial product terms generated, thereby enhancing efficiency. Meanwhile, the CLA is a high-speed adder architecture that minimizes propagation delays, enabling faster addition operations. Combining these two components within the MAC unit results in a powerful computational engine capable of executing multiply-accumulate operations with exceptional speed and precision. By leveraging the advantages of both the Booth multiplier and the CLA Adder, this MAC unit excels in handling complex calculations efficiently, making it indispensable in applications such as audio processing, image processing, and communications. This integration represents a strategic approach to addressing the performance constraints of conventional MAC units, offering a compelling solution for achieving optimal speed, accuracy, and efficiency in VLSI designs. In essence, the utilization of a Booth multiplier and a CLA Adder within the MAC unit heralds a new era of computational prowess, promising advancements in various domains where rapid and precise calculations are paramount.

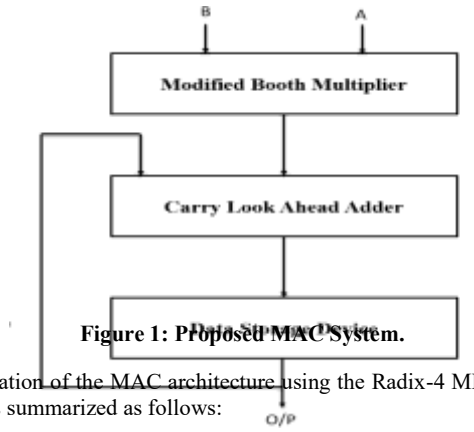

Figure 1: Proposed MAC System.

The operation of the MAC architecture using the Radix-4 MBM and EC-CLA was summarized as follows:

Input: The multiplier and multiplicand are inputted into the MAC module.

Radix-4 MBM Multiplication: The Radix-4 MBM performs the multiplication operation, generating partial products by multiplying the encoded segments of the multiplier with the multiplicand.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Partial Product Accumulation: The partial products are accumulated using the EC-CLA. The EC-CLA adds the partial products together while providing error correction capabilities to ensure data integrity.

Result: The final result of the MAC operation, i.e., the accumulated product of the multiplier and multiplicand, is obtained from the EC- CLA output.

Applications:

- **Automotive Control Systems**: The MAC unit's real-time capabilities are advantageous in automotive control systems.
- **Biomedical Signal Processing**: The designed MAC unit can find application in real-time biomedical signal processing, such as processing data from medical sensors and devices.
- **Real-Time Audio Processing**: The MAC unit was employed in audio processing applications, such as real- time audio filtering, equalization, and convolution.

Advantages:

- **Optimized Power Consumption:** The design includes optimization techniques that lead to more efficient power utilization. This is beneficial in applications with stringent power constraints, including portable devices and energy-conscious systems.
- **Ease of Integration into VLSI Systems:** The design's modular nature and compatibility with standard VLSI design practices make it easier to integrate into larger VLSI systems. This simplifies the overall design process and facilitates its adoption in various applications.
- **Error Resilience:** The self-error correction feature ensures that inaccuracies introduced during computation are identified and rectified, enhancing the overall resilience of the Multiply-Accumulate unit. This is crucial in applications where precise calculations are paramount.
- **Improved Accuracy in Results:** The incorporation of self-error correction mechanisms contributes to higher accuracy in the results produced by the MAC unit. This is particularly advantageous in scenarios where precision is critical, such as scientific simulations and financial modeling.

4. EXPERIMENTAL ANALYSIS

Figure 2 shows the existing simulation results for N=32 bit. Here, 'a' and 'b' are the inputs, 'en' is the enabled signal and should always be in a active state and 'clk' represents clock cycle.

Figure 2: Proposed Simulation Result

Figure 3 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. 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

Figure 3: Proposed area

Figure 4 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.

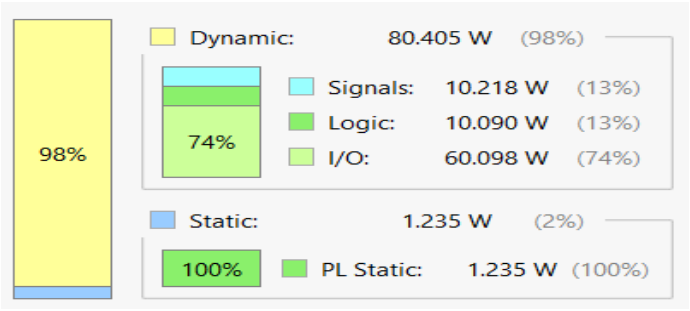

Figure 4: Proposed power

Figure 5 shows proposed setup delay for N=32. Here, Total delay is 23.705 ns, maximum Logic Delay is 7.669 ns and maximum Net delay is 16.036 ns..

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	20	15	33	a[16]	out_reg[61]/D	23.705	7.669	16.036
Path 2	∞	19	14	33	a[16]	out_reg[63]/D	23.690	7.575	16.115
Path 3	∞	19	14	33	a[16]	out_reg[62]/D	23.616	7.501	16.115
Path 4	∞	20	15	33	a[16]	out_reg[60]/D	23.565	7.529	16.036
Path 5	∞	17	12	33	a[16]	out_reg[59]/D	23.368	7.125	16.244
Path 6	∞	20	14	34	a[6]	out_reg[57]/D	23.317	8.547	14.770
Path 7	∞	17	12	33	a[16]	out_reg[58]/D	23.294	7.051	16.244
Path 8	∞	20	14	34	a[6]	out_reg[56]/D	23.177	8.407	14.770
Path 9	∞	19	13	34	a[6]	out_reg[53]/D	23.064	8.294	14.770
Path 10	∞	19	13	34	a[6]	out_reg[55]/D	23.044	8.274	14.770

Figure 5: Proposed setup delay

Figure 6 shows proposed setup delay for N=32. Here, Total delay is 0.589 ns, maximum Logic Delay is 0.345 ns and maximum Net delayis 0.244 ns.

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 11	∞	3	1	2	out_reg[27]/C	out_reg[27]/D	0.589	0.345	0.244
Path 12	∞	3	1	2	out_reg[35]/C	out_reg[35]/D	0.589	0.345	0.244
Path 13	∞	3	1	2	out_reg[43]/C	out_reg[43]/D	0.589	0.345	0.244
Path 14	∞	3	1	2	out_reg[47]/C	out_reg[47]/D	0.589	0.345	0.244
Path 15	∞	3	1	2	out_reg[11]/C	out_reg[11]/D	0.590	0.345	0.245
Path 16	∞	3	1	2	out_reg[15]/C	out_reg[15]/D	0.590	0.345	0.245
Path 17	∞	3	1	2	out_reg[19]/C	out_reg[19]/D	0.590	0.345	0.245
Path 18	∞	3	1	2	out_reg[23]/C	out_reg[23]/D	0.590	0.345	0.245
Path 19	∞	3	1	2	out_reg[31]/C	out_reg[31]/D	0.590	0.345	0.245
Path 20	∞	3	1	2	out_reg[39]/C	out_reg[39]/D	0.590	0.345	0.245

Figure 6: Proposed hold delay

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

5. CONCLUSION

The proposed 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 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.

So, 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.

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, Soa 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 Transactionson 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.

[11]. Noh, Seock-Hwan, Jahyun Koo, Seunghyun Lee, Jongse Park, and Jaeha Kung. "FlexBlock: A flexible DNN training accelerator with multi-mode block floating point support." *IEEE Transactions on Computers* (2023).

[12]. Kushwaha, Dinesh, Rajat Kohli, Jwalant Mishra, Rajiv V. Joshi, S. Dasgupta, and Anand Bulusu. "A Fully Differential 4-Bit Analog Compute-In-Memory Architecture for Inference Application." In *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 1-5. IEEE, 2023.

[13]. Wang, Chia-Chun, Yun-Chen Lo, Jun-Shen Wu, Yu-Chih Tsai, Chia-Cheng Chang, Tsen-Wei Hsu, Min-Wei Chu, Chuan-Yao Lai, and Ren-Shuo Liu. "Exploiting and Enhancing Computation Latency Variability for High- Performance Time-Domain Computing-in-Memory Neural Network Accelerators." In *2023 IEEE 41st International Conference on Computer Design (ICCD)*, pp. 515-522. IEEE, 2023.