International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

IJIEMR Transactions, online available on 31th Feb 2021. Link

https://ijiemr.org/downloads.php?vol=Volume-10&issue= Issue02

DOI:10.48047/IJIEMR/V10/ ISSUE02/24

Title: " OPTIMIZED MULTIPLY-ACCUMULATE UNIT DESIGN FEATURING SELF-ERROR CORRECTION AND ACCUMULATION MECHANISM"

Volume 10, ISSUE 02, Pages: 132- 140

Paper Authors
Saresh Kumar Ellamla, Kunsoth Umarani, Kaki Ajay

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

OPTIMIZED MULTIPLY-ACCUMULATE UNIT DESIGN FEATURING SELF-ERROR CORRECTION AND ACCUMULATION MECHANISM

Saresh Kumar Ellamla, Kunsoth Umarani, Kaki Ajay

Department of Electronics and Communication Engineering, Sree Dattha Group of Institutions, Sheriguda, Hyderabad, Telangana.

ABSTRACT

The design of Multiply-Accumulate (MAC) units is an essential component in the area of digital signal processing, since it is responsible for ensuring that high-performance computations are carried out. Applications of the MAC design can be found in a variety of industries, including digital signal processing, communications, and artificial intelligence. In these areas, MAC operations are essential for the efficient computing of convolutional neural networks, digital filters, and other signal processing tasks. At the moment, MAC units typically consist of separate adders and multipliers, which results in an increase in the complexity of the hardware and the amount of power that is consumed. The currently available systems frequently have difficulty finding a happy medium between speed and the utilization of resources. As a result of the independent implementation of adders and multipliers, traditional MAC architectures involve the utilization of redundant hardware. Adders and multipliers that are kept separate from one another contribute to higher power consumption, which in turn limits energy efficiency. As a result of its architecture, the currently available systems could experience difficulties in terms of scalability, particularly when attempting to achieve high-performance computing. This study presents a novel approach to the design of MAC units by utilizing unified adders and multipliers. The objective of this work is to improve both the speed of the system and its consumption of resources. Integrated unified adders and multipliers are incorporated into the suggested technique, which optimizes the MAC unit for increased speed and efficiency improvements. Through the employment of a unified architecture, the design brings about a reduction in power consumption, a reduction in redundancy, and an enhancement of resource utilization.

Keywords: Multiply-Accumulate Unit, Advanced Multipliers, Parallel Adders, Digital filters.

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. 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 must 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.

2. LITERATURE SURVEY

Di Meo, et.al [1] investigated a MAC unit which computed $Y = A \times B + C$ 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. 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. 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.

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. 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) scaling 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. 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.

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. Wang, et.al [8] presented a spin transfer torque magnetic random-access memory (STT-MRAM) in-memory multiplication structure based on enhanced readout margin through capacitor discharge and a positive feedback structure. On this basis, a computing-in-memory (CIM) macro for MAC operations was designed.

Jing, et.al [9] proposed a new accelerator design that leveraged bit sparsity in both weights and activations to improve performance. To harness the bit sparsity, they first proposed dynamically detecting the zero bits in activations and substituting the MAC units with bit-wise shift-and-accumulate units to sustain the computing parallelism. To avoid the random number and position of the zero bits, activation grouping and synchronization were proposed to dynamically balance the bit-wise workload. Antolini, et.al [10] presented a combined hardware and software solution to mitigate the impact of PCM non-idealities. The drift of PCM cell conductance was compensated at the circuit level through the introduction of a conductance ratio at the core of the MVM computation. A model of the behaviour of PCM cells was employed to develop a device-aware training for DNNs, and the accuracy was estimated in a CIFAR-10 classification task. Noh, et.al [11] presented FlexBlock, a DNN training accelerator with three BFP modes, possibly different among activation, weight, and gradient tensors. By configuring FlexBlock to a lower BFP precision, the number of MACs handled by the core increased by up to 4× in 8-bit mode or 16× in 4-bit mode compared to 16-bit mode.

Kushwaha, et.al [12] proposed a method that achieved a high signal margin for the 4-bit CIM architecture due to fully differential voltage changes on read bit-lines (RBL/RBLBs). The signal margin achieved for 4-bit MAC operation was 32 mV, which was 1.14×, 5.82×, and 10.24× higher than the state-of-the-art. The proposed scheme was robust against process, voltage, and temperature (PVT) variations. Wang, et.al [13] presented an architecture for a time-domain CIM-based neural network accelerator that leveraged the varying output time of the TDC. They introduced an early-termination scheme for time-domain CIM, which dynamically determined the length of the CIM clock period by deriving the maximum possible MAC value based on the current input. This approach reduced computation time for low-MAC results. Laxman, et.al [14] developed a low-power MAC unit and executed utilizing a hybrid logic technique to achieve power efficiency. The MAC unit was purposefully designed with suitable geometries to provide optimized power, area, and delay characteristics. The delay in the MAC unit was estimated based on data flow analysis between the MAC blocks, explicitly focusing on low-power concerns. Vaithiyanathan, et.al [15] presented architectures for single-channel and multichannel FIR filters employing the Time-division multiplexing (TDM) scheme. The studied architecture was associated with one multiplication and addition unit to handle a wide range of channels and filter taps to have efficient resource utilization. Further, accumulator-based Radix-4 multiplier, shift and add multiplication, and parallel pipelined multiplication operations involved in the architectures effectively utilized the resources to a considerable extent.

3. PROPOSED METHODOLOGY

The Radix-4 Modified Booth Multiplier (MBM) is a key component in the MAC architecture, particularly for efficiently computing the product of the multiplier and multiplicand. This optimized hardware module implements the Modified Booth algorithm with a radix of 4, which significantly reduces the number of partial product rows needed for multiplication. This reduction in partial product rows leads to improved efficiency and reduced hardware complexity, making it an attractive choice for various applications.

After the multiplication operation is completed by the Radix-4 MBM, the resulting partial products are accumulated to obtain the final product in the MAC architecture. This accumulation process is facilitated by an EC-CLA, which is a specialized adder circuit designed to efficiently perform multi-bit addition operations while providing error correction capabilities. The EC-CLA operates in several stages. Initially, the partial products generated by the Radix-4 MBM are inputted into the EC-CLA module. The EC-CLA utilizes a carry lookahead architecture, which allows for the pre-computation of carry signals for each pair of bits in parallel. This parallel computation enables faster addition of multi-bit numbers compared to traditional ripple carry adders, enhancing the overall performance of the MAC operation. In the MAC architecture, data storage units play a crucial role in maintaining the coherence and efficiency of data movement during computation. These units, typically consisting of registers or memory elements, are essential for storing intermediate results and operands at various stages of the MAC operation.

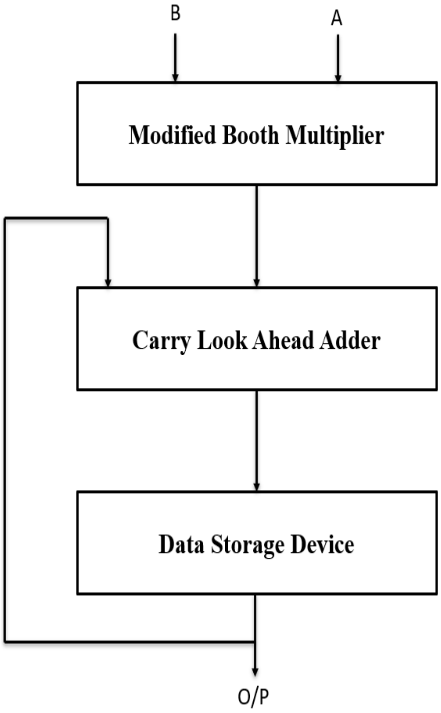


Figure 1. Proposed block diagram.

The operation of the MAC architecture using the Radix-4 MBM and EC-CLA is shown in Figure 1, which can be summarized as follows:

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

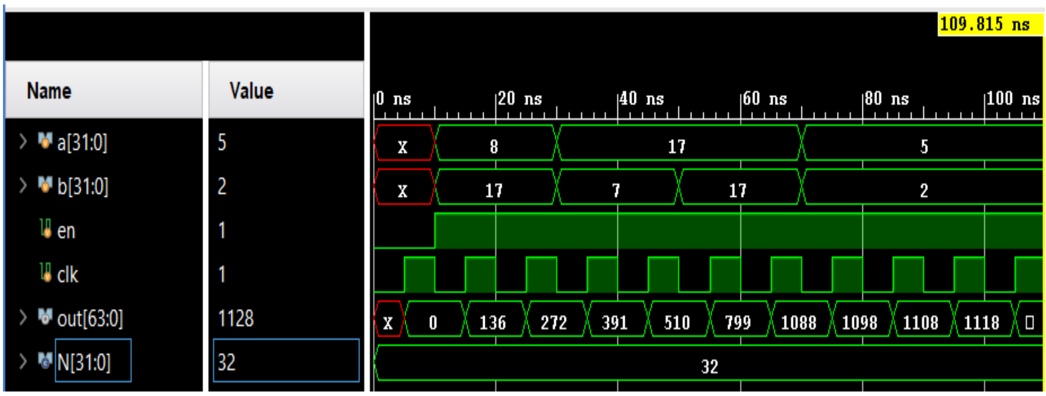
Radix-4 MBM Multiplication: The Radix-4 MBM performs the multiplication operation, generating partial products by multiplying the encoded segments of the multiplier with the multiplicand.

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 result of the MAC operation, i.e., the accumulated product of the multiplier and multiplicand, is obtained from the EC-CLA output. The proposed MAC architecture offers a balance of efficiency, reliability, and scalability. It efficiently handles both positive and negative numbers, offers scalability to support different radices, and ensures reliable computation through error correction mechanisms. The architecture's ability to perform complex arithmetic operations with minimal latency and maximum efficiency makes it well-suited for a wide range of high-performance computing applications.

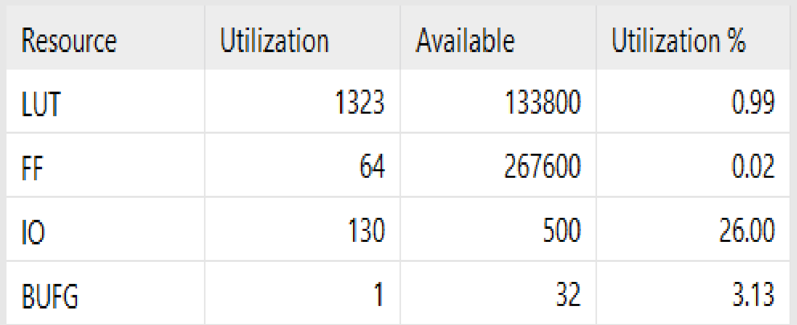
4. RESULTS AND DISCUSSION

Figure 2 illustrates the outcomes of simulating the MAC unit implementations. Figure 3 provides a summary of the design characteristics of the MAC unit implementations. Figure 4 presents a summary of the power consumption metrics of the MAC unit implementations. It may include information on static power (power consumed when the unit is idle) and dynamic power (power consumed during operation), offering insights into the energy efficiency of each design. Figure 5 depicts a summary of the time-related metrics of the MAC unit implementations. It could include metrics such as processing time per operation or overall processing time for a given dataset, indicating the computational efficiency and latency of each design. Table 1 provides a comparison between the different MAC unit implementations across various metrics or parameters.



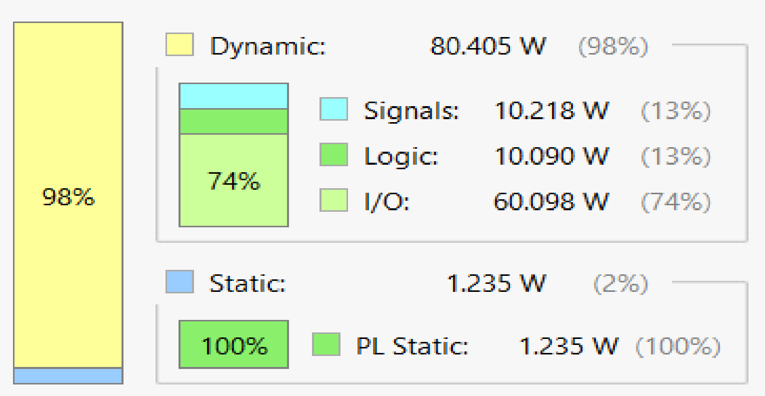
Name	Value
a[31:0]	5
b[31:0]	2
en	1
clk	1
out[63:0]	1128
N[31:0]	32

Figure 2: Simulation Results.



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: Design summary.



Dynamic: 80.405 W (98%)

Signals: 10.218 W (13%)

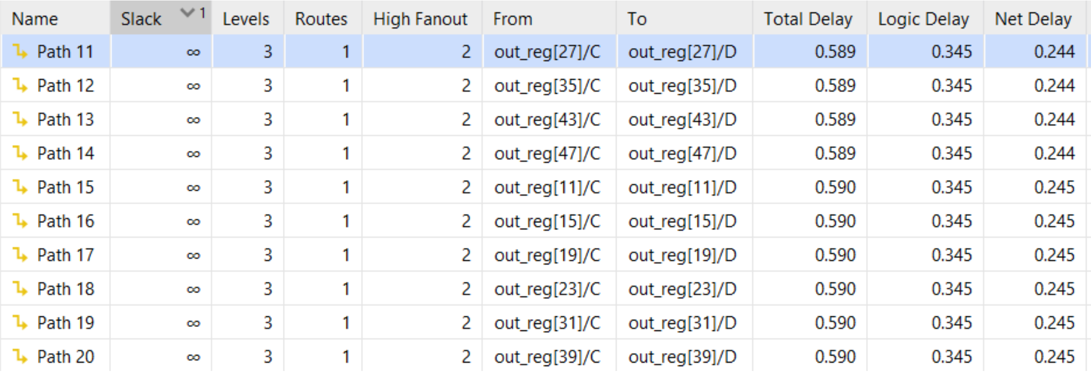
Logic: 10.090 W (13%)

I/O: 60.098 W (74%)

Static: 1.235 W (2%)

PL Static: 1.235 W (100%)

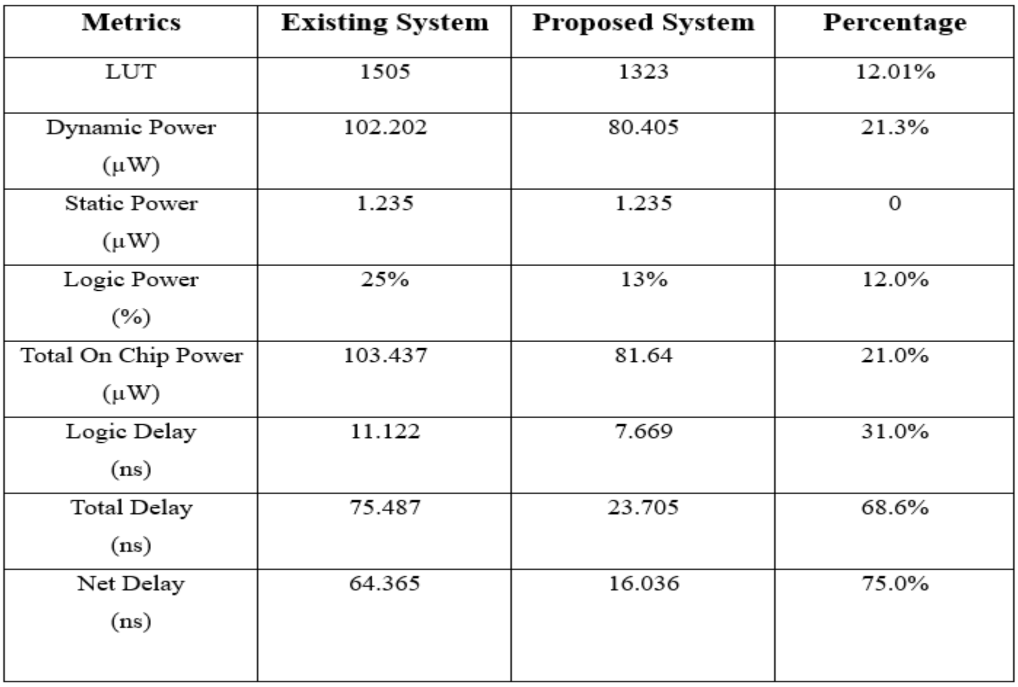
Figure 4: Power summary.



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 5: Time summary.

Table 1: Comparison Table



Metrics	Existing System	Proposed System	Percentage
LUT	1505	1323	12.01%
Dynamic Power (μW)	102.202	80.405	21.3%
Static Power (μW)	1.235	1.235	0
Logic Power (%)	25%	13%	12.0%
Total On Chip Power (μW)	103.437	81.64	21.0%
Logic Delay (ns)	11.122	7.669	31.0%
Total Delay (ns)	75.487	23.705	68.6%
Net Delay (ns)	64.365	16.036	75.0%

5. 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 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.

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.

[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.

[14]. Laxman, Amgoth, N. Siva Sankara Reddy, and B. Rajendra Naik. "Design and implementation of hybrid logic-based MAC unit using 45 nm technology." *e-Prime-Advances in Electrical Engineering, Electronics and Energy* 6 (2023): 100317.

[15]. Vaithiyanathan, Dhandapani, Britto Pari James, and Karuthapandian Mariammal. "Comparative Study of Single MAC FIR Filter Architectures with Different Multiplication Techniques." In *2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, pp. 01-10. IEEE, 2023.