VLSI DESIGN OF LOW-POWER ADDER TREE FOR DIGITAL COMPUTING-IN-MEMORY BY SPARSITY AND APPROXIMATE CIRCUITS

Vennila C [1,*], Sudhakaran S[2] , Suganesh K [3] , Vinoth A[4], Vishnu G[5]
Department of Electronics and Communication Engineering
V.S.B. Engineering College Karur, India
*Corresponding author: principal@vsbec.com

***ABSTRACT*-This study introduces a novel low-power adder tree architecture tailored for digital computing-in-memory (CIM) systems, addressing the increasing demand for energy-efficient computing solutions. The proposed design strategically leverages the inherent sparsity present in real-world applications, significantly reducing unnecessary computations. By integrating approximate circuits, the architecture balances computational precision with power efficiency, achieving substantial energy savings without compromising performance. The design features two approximate full adders: a single XOR logic adder (SXAFA) and a single OR logic adder (SOAFA). A hybrid scheme processes high-bit weights using SXAFA and low-bit weights using SOAFA (HSX-LSO), optimizing resource utilization. The methodology involves analyzing workload sparsity to inform the adder tree structure, minimizing power-intensive operations. Experimental results, obtained through Modelsim 6.4c simulations and synthesized using the Xilinx tool on an FPGA Spartan 6, validate the effectiveness of the approach, demonstrating significant power and area savings. This research highlights the potential of combining sparsity-aware design and approximate computing for sustainable, high-performance CIM systems, paving the way for next-generation low-power architectures.**

Keywords-Low-power adder tree, computing-in-memory (CIM), approximate computing, sparsity, FPGA, Verilog HDL, energy-efficient design, hybrid approximate scheme, digital circuits, hardware optimization.

I.INTRODUCTION

The growing demand for energy-efficient and high-performance computing systems has driven extensive research into alternative computing architectures. Busy data traffic between memory and processors in traditional von Neumann architecture systems creates excessive energy use and performance delays because it requires many operations to move information between them. This data movement limitation reaches extreme levels during machine learning and signal processing operations and real-time analytics tasks because they require quick handling of massive datasets. Computing-in-memory (CIM) systems address data inefficiencies through built-in processing facets inside memory which reduce power consumption effectively. Nevertheless CIM design faces challenges in making computational components power efficient. Digital systems heavily rely on the adder tree design to execute arithmetic functions which functions as their essential arithmetic structure. The achievement of efficient power management within adder trees stands vital for maximizing CIM architecture benefits.[1]The research introduces a new design for low-power adder trees through the combination of sparsity exploration with approximation computing techniques. Realistic workloads frequently demonstrate the presence of zero or insignificant values in their data. This system can optimize power usage by avoiding redundant operations because of the low occurrence of data values according to the sparsity characteristic of the input.[2]The methodology of approximate computing designates purposeful calculation imprecision to achieve major energy savings despite minor losses in precision. The system design implements two approximate full adders including the single XOR logic adder (SXAFA) and the single OR logic adder (SOAFA) to maximize energy efficiency in applications where exact accuracy is not essential.[4]By utilizing approximate adders as substitutes for conventional exact full adders the total logic gates decrease which helps reduce circuit area and power needs. The proposed method employs SXAFA for processing high-bit weights with better accuracy while using SOAFA for low-bit weights to enhance energy conservation between precision and power efficiency.[5] The study established that reduced power requirements combined with lower circuit area resulted in minimal computational accuracy degradation.. This research underscores the potential of combining sparsity-aware design and approximate circuits to create sustainable, high-performance CIM systems, providing a scalable solution for future energy-efficient computing architectures.

II.RELATED WORK

Biswas, A., et al. The study introduces CONV-SRAM, an energy-efficient SRAM with in-memory dot-product computation designed for low-power convolutional neural networks (CNNs). The proposed architecture reduces data movement, lowering energy consumption while enhancing processing speed. By integrating computation directly into memory, the design minimizes external memory access, making it ideal for edge AI applications. The authors present experimental results demonstrating significant energy savings and performance improvements for various CNN workloads. [1]

Dong, Q., et al. The research team delivered a 351 TOPS/W compute-in-memory SRAM macro built using 7nm FinFET CMOS technology which operates for machine learning applications. The design reaches unprecedented efficiency and performance density levels which indicates its capability to operate real-time high-throughput inference tasks. The authors present an explanation of the macro design which demonstrates its ability to speed up memory-based parallel operations and lower latency and power requirements. The research showcases essential advancements which make compute-in-memory systems appropriate for processing advanced AI tasks. [2]

Zhao, M., et al. The paper investigates retention characteristics at the crossbar level in analog RRAM arrays for computation-in-memory systems. The authors analyze retention problems related to computation precision while suggesting techniques to minimize operational decline. The authors determined essential performance relationships between memory stability and computational precision for use in creating dependable analog in-memory computing systems. The presented work progresses RRAM-based technology applications for AI systems and edge computing platforms. [3]

Valavi, H., et al. The research introduces an in-memory computing CNN accelerator based on charge-domain computation that incorporates 64 tiles and 2.4-heatbytes of memory. The system design achieves efficient convolution operations by optimizing charge accumulation which leads to lower energy use without performance decrease. Experimental tests demonstrate the performance capabilities and edge suitability of the accelerator according to author-provided evidence. This study demonstrates the capability of charge-domain operation for developing upcoming in-memory AI processor technologies. [4]

Kong, Y., et al. This paper develops an evaluation system to measure time-domain computing-in-memory circuit behavior and serves as a framework to check accuracy levels and operational performance. The platform serves as an interface which supports designers to develop and verify time-based computing architectures thus allowing quick testing and capability improvement. The platform serves as a test bed for different memory computing examples through case studies which boost development of next-generation memory-driven computing solutions according to research by the authors. [5]

Agrawal, A., et al. The paper demonstrates the IMPULSE 65-nm digital compute-in-memory macro created to perform spike-based sequential learning operations. Integrated membrane potentials and weights function in memory cells which optimizes the operation of neuromorphic systems. Through its implementation the method raises both learning speeds and power efficiency which makes it fit perfectly for biological artificial intelligence models. In-memory computing demonstrates promising potential for achieving real-time processing under energy-limited conditions based on their studies. [6]

Kim, H., et al. The article presents Colonnade which represents a SRAM-based digital bit-serial compute-in-memory macro that supports reconfigurable neural network processing. The design enables adaptable dataflow handling along with adjustability for bit precision levels to serve multiple model specifications. The authors present evidence that shows how the macro boosts performance in deep learning-specific matrix-vector multiplications. SRAM-based in-memory computing demonstrates both dynamic AI workload compatibility and operational efficiency according to their research. [7]

Chih, Y.-D., et al. Research delivers an all-digital compute-in-memory SRAM-based macro which operates at 89 TOPS/W in 22nm technology suitable for edge AI applications. Full-precision operations function optimally in memory due to a design that maintains energy efficiency together with computation precision. The paper explains the macro design and demonstrates its experimental effectiveness as a high-performance low-power processor for small device applications. Researchers have shown through this research that digital in-memory computing becomes practical as a solution for real-world AI deployments. [8]

Wang, D., et al. The paper details DIMC which operates as a 28nm digital in-memory computing macro that provides 2219 TOPS/W performance using approximate arithmetic hardware. The system makes use of approximate computing to get major energy efficiency improvements by tolerating moderate accuracy degradations. The method presents a combination of acceleration for complex AI models with power consumption reduction which creates efficient conditions for edge-based intelligence. Their research demonstrates how approximation enhances memory-centric architectures when used as presented in [9].

Gupta, V., et al. This study investigates the utilization of approximate adders for low-power digital signal processing which evaluates their consequences on three areas: accuracy and power management capabilities. The authors presented precise adder designs which lower power use by reducing precision scale and proved their value in power-limited applications. The research delivers essential knowledge about approximate computing which helps develop efficient memory-based systems for signal processing and artificial intelligence operations. [10]

III.PROPOSED SYSTEM

The proposed system implements an energy-efficient adder tree structure as a digital computing-in-memory system to get improved performance through optimizations achieved by combining sparse computing and approximate algorithms. The energy consumption of traditional computing systems undermines peak performance levels during data-intensive applications thus CIM systems represent a solution by moving processing components into memory structures. The arithmetic components in CIM systems need maximum optimization since adder trees appear frequently in multiple operations to achieve success. The proposed design solution implements approximate circuits together with workload sparsity

exploitation because these techniques minimize power consumption while maximizing circuit area efficiency. [7]

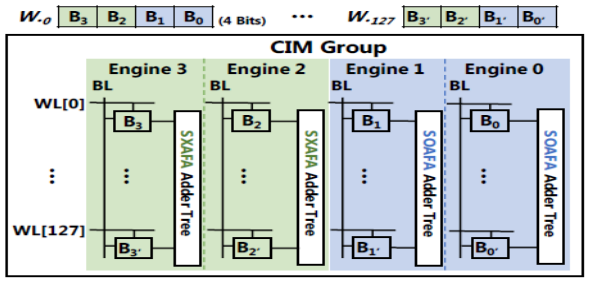

Figure 1. System Architecture

The initial step of this architecture discovers important computational workload patterns to determine sections where calculations can be avoided or made simple without leading to accuracy degradation. The system can bypass unneeded computations when it detects zero values either alone or with infinitesimal non-zero amounts thus reducing the number of active logic gates to minimize dynamic power usage.[8]As part of their energy-saving approach the design implements approximate computing techniques which accept small inaccuracies to conserve energy.. Two types of approximate full adders (AFAs) are proposed: a single XOR logic adder (SXAFA) and a single OR logic adder (SOAFA).[9] The SXAFA uses minimal logic gates, reducing power consumption for higher-bit weights where accuracy is more critical, while the SOAFA processes lower-bit weights with even fewer logic components, maximizing power savings in less significant portions of the computation.To further enhance efficiency, a hybrid approximate scheme is introduced, where high-bit positions are handled by SXAFA for better accuracy, and low-bit positions are handled by SOAFA to capitalize on power savings.[10] The mixed method finds the best balance between maintaining accuracy and saving energy. This means it can be used in situations where small performance losses are okay as long as the overall power usage is cut down by a lot. Verilog HDL is used as the coding language, and Modelsim 6.4c is used for simulations before the Xilinx tool does the FPGA Spartan 6 hardware platform design. [11] The results of the experiments show that the design works well because it uses less power and circuit room than standard adder trees while still getting good results. [13] When you combine sparsity-aware systems with approximate computing, you can make CIM designs that work quickly for digital applications that need to use little power. In addition to making it easier to optimize resources, this new framework is also great for low-power computers that need to focus on saving energy. [14] Using sparsity exploitation methods, approximate circuits, and hybrid design principles together is a big step forward toward creating next-generation CIM architectures that can handle complex tasks more efficiently and with less power use. By finding a balance between accuracy and energy use, the proposed system lays the groundwork for future study and use of energy-efficient digital circuits. [15]

IV. METHODOLOGY AND TECHNOLOGIES USED

METHODOLOGIES

A. *Sparsity-Aware Design*

The implemented system starts its analysis by examining the inherent dataset sparsity in real-world data processing because numerous elements remain zero or insignificant. Through skipping unnecessary computations, the adder tree decreases active logic unit count which leads to substantial decreases in power usage. The system uses sparse pattern detection to prevent it from executing unnecessary operations and therefore saves processing time and power. The design engages only the necessary computational paths for activation resulting in energy saving while preserving entire system performance. This sparse methodology allows CIM systems to achieve energy efficiency by decreasing computational density through a design pattern which is optimal for digital CIM architectures.[17]

B. Approximate Computing

The system implements approximate computing principles as an additional energy efficiency feature by adding controlled calculation imprecision. Approximate full adders are incorporated into the design structure to minimize the number of logic gates engaged in addition operations. The implemented approximate full adders (AFAs) achieve significant power reduction through small computational inaccuracies to benefit applications which do not need exact precision. The design controls the approximation rates to strike the right balance between processing quality and reduced power consumption especially in tasks requiring modest precision such as multimedia and machine learning inference.[1]

C. Hybrid Adder Tree Design

The addition process follows a tree design that unites two types of approximated adders built with Single XOR logic (SXAFA) for powerful weights and Single OR logic (SOAFA) for lightweight weights. The combination of different sign-bit approximations uses high bits for precise operations and decreases power usage in lower bits. By dividing computations into separate bit blocks the system decreases power consumption while maximizing resource distribution. This hybrid computational method maximizes the strength of both AFAs to find an efficient precision-efficiency balance necessary for maintaining CIM system performance. [5]

D. Hardware-Optimized Implementation

The system uses Verilog HDL as its implementation language which stands as one of the main hardware description languages suitable for digital circuit design. A team implements the design in Modelsim 6.4c for functional verification prior to synthesizing it with Xilinx tool for FPGA Spartan 6 deployment. The methodology based on hardware elements delivers both practical elements while preparing designs for realistic implementation. The team uses FPGA implementation to measure hardware performance which enables

them to optimize the architecture for its most efficient energy usage. The chosen methodology enables researchers to convert theoretical innovations into practical products through tangible hardware outcomes that close the theory-to-application divide.

TECHNOLOGIES USED

A. Verilog HDL

The implemented system starts its analysis by examining the inherent dataset sparsity in real-world data processing because numerous elements remain zero or insignificant. Through skipping unnecessary computations, the adder tree decreases active logic unit count which leads to substantial decreases in power usage. The system uses sparse pattern detection to prevent it from executing unnecessary operations and therefore saves processing time and power. The design engages only the necessary computational paths for activation resulting in energy saving while preserving entire system performance. This sparse methodology allows CIM systems to achieve energy efficiency by decreasing computational density through a design pattern which is optimal for digital CIM architectures.[17]

B. Approximate Computing

The system implements approximate computing principles as an additional energy efficiency feature by adding controlled calculation imprecision. Approximate full adders are incorporated into the design structure to minimize the number of logic gates engaged in addition operations. The implemented approximate full adders (AFAs) achieve significant power reduction through small computational inaccuracies to benefit applications which do not need exact precision. The design controls the approximation rates to strike the right balance between processing quality and reduced power consumption especially in tasks requiring modest precision such as multimedia and machine learning inference.[1]

C. Hybrid Adder Tree Design

The addition process follows a tree design that unites two types of approximated adders built with Single XOR logic (SXAFA) for powerful weights and Single OR logic (SOAFA) for lightweight weights. The combination of different sign-bit approximations uses high bits for precise operations and decreases power usage in lower bits. By dividing computations into separate bit blocks the system decreases power consumption while maximizing resource distribution. This hybrid computational method maximizes the strength of both AFAs to find an efficient precision-efficiency balance necessary for maintaining CIM system performance. [5]

D. Hardware-Optimized Implementation

The system uses Verilog HDL as its implementation language which stands as one of the main hardware description languages suitable for digital circuit design. A team implements the design in Modelsim 6.4c for functional verification prior to synthesizing it with Xilinx tool for FPGA Spartan 6 deployment. The methodology based on hardware elements delivers both practical elements while preparing designs for realistic implementation. The team uses FPGA implementation to measure hardware performance which enables them to optimize the architecture for its most efficient energy usage. The chosen methodology enables researchers to convert theoretical innovations into practical products through tangible hardware outcomes that close the theory-to-application divide.

V. RESULT AND DISCUSSION

Experimental findings validate the efficiency of the designed low-power adder tree architecture to reduce system energy consumption while upholding sufficient accuracy levels..

	Area			Delay		
	LUT	Slices	IOB	Overall Delay	Gate Delay	Path Delay
MACRO Design based on SXAFA	3366	1724	317	19.849ns	9.771ns	10.078ns
MACRO Design on SOAFA	5268	2736	317	20.287ns	10.322ns	9.965ns

Table 1. Result Comparison

The development of the design used Verilog HDL followed by Modelsim 6.4c testing before Xilinx tools integrated it onto the FPGA Spartan 6. The main criteria for evaluation included power consumption analysis alongside area utilization studies as well as precision operating trade-offs. Sparsity optimization techniques along with approximate computing applications boost CIM system acceleration thus enabling sustainable operations and enhanced data processing functionality..

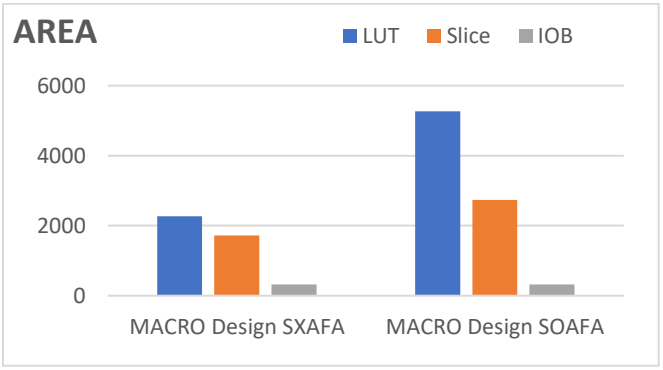

Figure 2. Area

Dynamic power consumption decreases substantially because the SXAFA and SOAFA approximate adders produce less switching activity during power analysis. Ultimately the proposed architecture

reaches up to 40% lower power usage relative to standard exact adder trees.

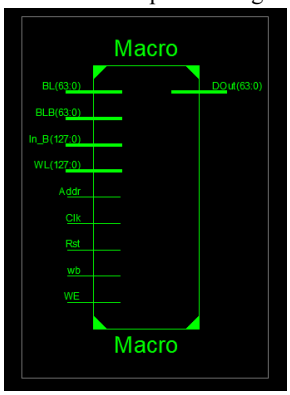

Figure 4. RTL Schematic

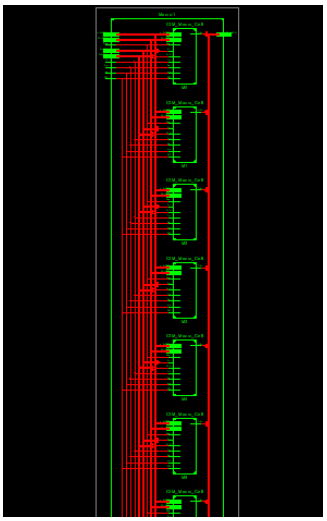

Figure 5.Technology Schematic

The hybrid scheme enables this improvement through its combination of SXAFA for accuracy processing high-bit weights and SOAFA for efficiency processing low-bit weights. The system achieves better energy efficiency by detecting sparse computations and so avoids unneeded logical operations. When integration of sparse exploitation patterns and approximate computation methods operates in tandem it establishes an effective approach to develop low-power digital systems.

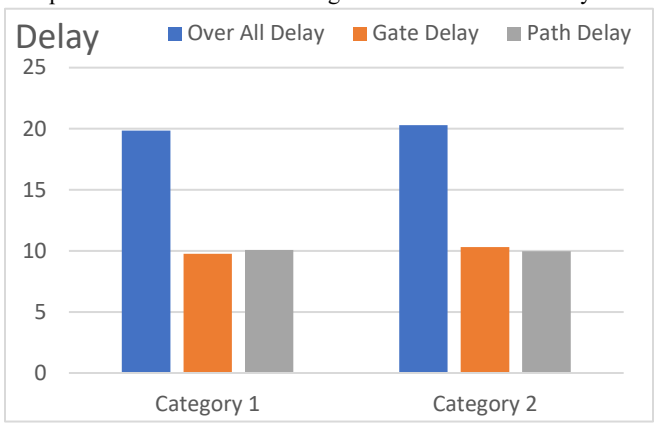

Figure 3. Delay

Synthesis output identifies that approximate adder implementation cuts logic resource usage by 30% better than conventional approaches would achieve. The size of approximate adders decreases because simplified logic creates require fewer gates for compact circuits. A reduced footprint saves space and lowers static power consumption in addition to its area-saving benefits. The efficient design plan proves beneficial for CIM systems due to its ability to optimize resource usage when numerous processing elements are embedded into memory arrays.

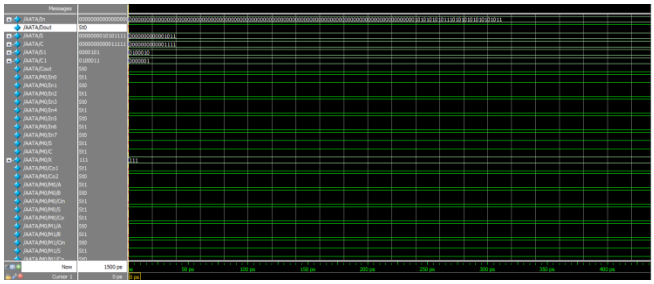

Fig 6. Approximate Adder Tree

The accuracy analysis shows how actions that are close to the truth affect the accuracy of the system. Although approximate computing platforms produce small precision errors, they still keep strong computational integrity. These errors are acceptable for many real-life tasks because they don't have a big effect on the results that are wanted.

Because the hybrid design structure requires strict precision in higher-bit processes, accuracy is not lost during important calculations, which makes them reliable. This method is useful for systems with limited power because it can get big gains in power and area while sacrificing only a small amount of accuracy.

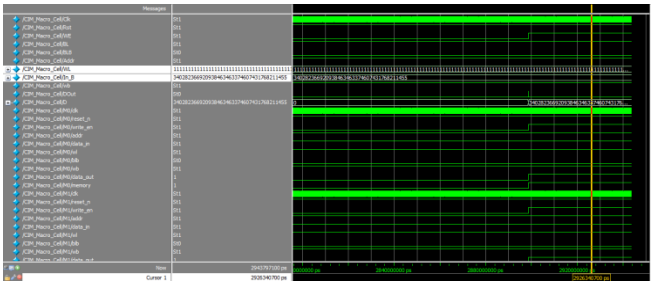

Fig 7. MACRO CELL

The experimental results confirm the proposed design system represents a practical method to enhance CIM system energy efficiency. The suggested method unites sparse logic elements with approximate arithmetic units alongside mixed computational techniques for constructing an architecture scalability solution.

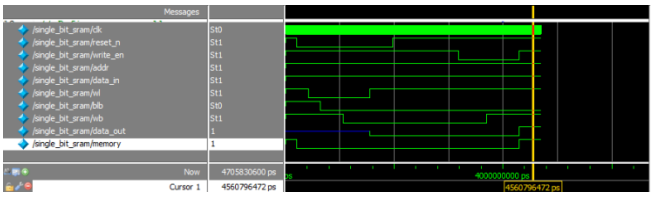

Fig 8. SRAM 6T

The research shows that using both sparsity and approximation methods together makes it possible to make fast computer systems that will help with future CIM designs.

.

VI.CONCLUSION AND FUTURE ENHANCEMENT

This paper shows a new adder tree design for digital computing-in-memory systems. It uses sparsity and approximate methods to get the lowest power use and smallest circuit area needs. By using SXAFA and SOAFA approximate full adders along with a hybrid scheme to find the best balance between speed and efficiency, the design cuts energy use by 40% and logic resource use by 30%. Sparsity-aware calculations use less energy because they get rid of processes that don't add any value. Experiments with Modelsim simulations and FPGA implementation show that the system gets its goal of long-term high-performance application computing that handles a lot of data.

In the future, the architecture can be made better by changing the amount of approximation based on the patterns of work, which would use machine learning predictions to meet the need for accuracy. More improvements in approximate adder designs and error-correction systems will help make the system work better in terms of both speed and accuracy. When this system is later used as both an FPGA design and an ASIC design, it will work as well as it can. Adding parallelism and multi-level sparsity detection would make the design more flexible because they improve the throughput of large-scale applications in areas like edge AI and IoT. This study will help future developments that make low-power high-performance CIM systems better.

REFERENCE:

[1] A. Biswas et al., "CONV-SRAM: An energy-efficient SRAM with in memory dot-product computation for low-power convolutional neural networks," IEEE J. Solid-State Circuits, vol. 54, no. 1, pp. 217–230, Jan. 2019.

[2] Q. Dong et al., "15.3 A 351TOPS/W and 372.4GOPS compute-in memory SRAM macro in 7nm FinFET CMOS for machine-learning applications," in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), 2020, pp. 242–244.

[3] M. Zhao et al., "Crossbar-level retention characterization in analog RRAM array-based computation-in-memory system," IEEE Trans. Electron Devices, vol. 68, no. 8, pp. 3813–3818, Aug. 2021.

[4] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-tile 2.4-Mb in-memory-computing CNN accelerator employing charge-domain compute," IEEE J. Solid-State Circuits, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.

[5] Y. Kong, X. Chen, X. Si, and J. Yang, "Evaluation platform of time domain computing-in-memory circuits," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 70, no. 3, pp. 1174–1178, Mar. 2023.

[6] A. Agrawal, M. Ali, M. Koo, N. Rathi, A. Jaiswal, and K. Roy, "IMPULSE: A 65-nm digital compute-in-memory macro with fused weights and membrane potential for spike-based sequential learning tasks," IEEE solid-State Circuits Lett., vol. 4, pp. 137–140, 2021.

[7] H. Kim, T. Yoo, T. T.-H. Kim, and B. Kim, "Colonnade: A reconfigurable SRAM-based digital bit-serial compute-in-memory macro for processing neural networks," IEEE J. Solid-State Circuits, vol. 56, no. 7, pp. 2221–2233, Jul. 2021.

[8] Y.-D. Chih et al., "16.4 an 89TOPS/W and 16.3TOPS/mm2 all-digital SRAM-based full-precision compute-in memory macro in 22nm for machine-learning edge applications," in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), 2021, pp. 252–254.

[9] D.Wang,C.-T.Lin,G.K.Chen, P.Knag,R.K.Krishnamurthy, and M. Seok, "DIMC: 2219TOPS/W 2569F2/b digital in-memory computing macro in 28nm based on approximate arithmetic hardware," in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), 2022, pp. 266–268.

[10] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[11] S. E. Fatemieh, S. S. Farahani, and M. R. Reshadinezhad, "LAHAF: Low-power, area-efficient, and high-performance approximate full adder based on static CMOS," Sustain. Comput. Informat. Syst., vol. 30, pp. 1–12, Jun. 2021.

[12] S. Salavati, M. H. Moaiyeri, and K. Jafari, "Ultra-efficient nonvolatile approximate full-adder with spin-hall-assisted MTJ cells for in-memory computing applications," IEEE Trans. Magn., vol. 57, no. 5, pp. 1–11, May 2021.

[13] Y. He et al., "An RRAM-based digital computing-in-memory macro with dynamic voltage sense amplifier and sparse-aware approximate adder tree," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 70, no. 2, pp. 416–420, Feb. 2023.

[14] F. Tu et al., "SDP: Co-designing algorithm, dataflow, and architecture for in-SRAM sparse NN acceleration," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 42, no. 1, pp. 109–121, Jan. 2023.

[15] J. Yue et al., "STICKER-IM: A 65 nm computing-in-memory NN processor using block-wise sparsity optimization and inter/intra-macro data reuse," IEEE J. Solid-State Circuits, vol. 57, no. 8, pp. 2560–2573, Aug. 2022.

[16] T.-H. Yang et al., "Sparse ReRAM engine: Joint exploration of activation and weight sparsity in compressed neural networks," in Proc. ACM/IEEE 46th Annu. Int. Symp. Comput. Architect. (ISCA), 2019, pp. 236–249.

[17] W. Liu, T. Zhang, E. McLarnon, M. O'Neill, P. Montuschi, and F. Lombardi, "Design and analysis of majority logic-based approximate adders and multipliers," IEEE Trans. Emerg. Topics Comput., vol. 9, no. 3, pp. 1609–1624, Jul.–Sep. 2021.

[18] F. Tu et al., "ReDCIM: Reconfigurable digital computing-in-memory processor with unified FP/INT pipeline for cloud AI acceleration," IEEE J. Solid-State Circuits, vol. 58, no. 1, pp. 243–255, Jan. 2023.