Proceedings of CONF-FMCE 2025 Symposium: Semantic Communication for Media Compression and Transmission
DOI: 10.54254/2755-2721/2025.GL25864

Computing-in-Memory Technology Review: From Circuit Design to Emerging Applications

Xiangde Li[1*†], Shixuan Wu[2†], Dian Jin[2]

[1]College of Integrated Circuit Science and Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China
[2]2011 College, Nanjing Tech University, Nanjing, China
**Corresponding Author. Email: b22030415@njupt.edu.cn*
[†]These authors contributed equally to this work and should be considered as co-first author.

Abstract. With the rapid development of artificial intelligence (AI) and big data applications, the memory wall bottleneck inherent in traditional von Neumann architectures has become increasingly prominent. As a promising solution, computing-in-memory (CIM) technology tightly integrates computing and storage units, significantly reducing the energy consumption and latency associated with frequent data movement between memory and processors. This paper reviews recent advances in CIM technology from three perspectives: circuit design, architectural optimization, and application innovation. Key techniques such as mixed-signal computing and heterogeneous memory integration are analyzed at the circuit level. The evolution from integer-only to floating-point (FP) and hybrid integer-floating-point (INT-FP) computing paradigms is discussed at the architectural level. Optimization strategies for emerging scenarios, such as on-chip training and simultaneous localization and mapping (SLAM) or neural radiance field (NeRF)-based environmental modelling, are explored at the application level. The findings suggest that CIM technology is evolving from domain-specific accelerators toward general-purpose computing platforms, with the potential to reshape next-generation intelligent computing architectures.

Keywords: Computing-in-memory, hybrid-domain computing, heterogeneous integration

1. Introduction

AI advancements demand new computing architectures. Current implementations use either cloud computing, offering high performance but facing latency and privacy issues, or edge computing, which improves responsiveness but requires extreme energy efficiency. The von Neumann architecture's processor-memory separation creates fundamental data transfer bottlenecks. With slowing semiconductor scaling, this limitation grows more severe. CIM technology solves this by embedding computation within memory arrays, transforming data processing paradigms.

Early CIM research focused on fundamental MACRO units using volatile (SRAM/DRAM) or non-volatile (RRAM/MRAM) memory to establish compute-memory integration feasibility. The field has since evolved to processor-level implementations combining optimized memory architectures with advanced computing logic, significantly enhancing practical applications.

DOI: 10.54254/2755-2721/2025.GL25864

Current CIM taxonomies based on memory types or computing paradigms inadequately represent recent advances. Modern designs employ hybrid computing and mixed-precision processing, while heterogeneous architectures integrate multiple memory technologies.

Therefore, this paper systematically reviews recent advances in CIM technology across three dimensions: circuit design, micro-architecture-level CIM macros, and applications, as shown in Fig. 1. Section II discusses circuit-level innovations, including hybrid-domain circuit designs and hybrid memory cells. Section III analyzes the architectural evolution from integer-only to FP and mixed INT-FP computing, along with emerging approximate and sparse computing techniques aimed at energy optimization. Section IV highlights application-level developments, focusing on AI workloads such as on-chip learning. Finally, Section V concludes this paper.

2. Circuit-level CIM design

Circuit-level innovations drive CIM's performance breakthroughs through two key approaches: hybrid computing circuits for superior energy-area efficiency and heterogeneous memory integration enabling high-density CIM macro units with low non-linearity.

2.1. CIM computing circuit

CIM computing circuits feature four principal implementations: current-, charge-, time-, and digital-based schemes. Early designs employed either purely analog or digital modalities. While analog CIM enables parallel matrix operations but suffers accuracy limitations, digital CIM ensures precision at the cost of energy and area efficiency.

2.1.1. Hybrid analog-digital computing circuit

Figure 1 shows that hybrid analog-digital CIM architectures optimally balance the analog's energy-efficient parallelism with the digital's computational precision, enabling high-performance neural network acceleration.

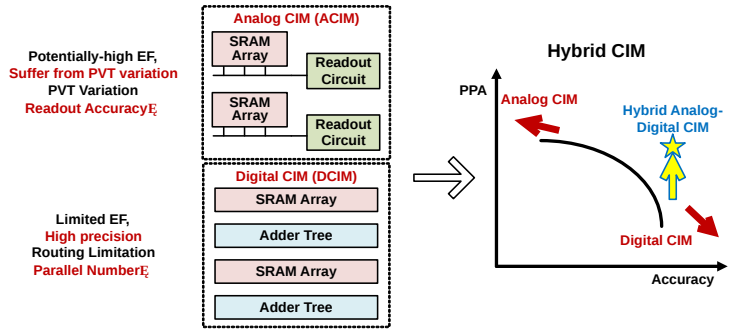

Figure 1. Analog, digital and hybrid CIM: performance comparison

Wu et al. [1] demonstrated a 22nm 832Kb hybrid-domain FP-SRAM CIM macro, which implements diagonal partial product partitioning with digital most significant bit (MSB) processing and analog least significant bit (LSB) accumulation. Jeong et al. [2] presented a 28nm CIM chip that harmonizes digital and analog computing through three key innovations: an optimized hybrid architecture, charge-recycling techniques, and streamlined adder design. This co-design approach simultaneously enhances energy efficiency and reduces area while maintaining computational throughput. Guo et al. [3] developed a novel lightning-patterned hybrid computing architecture that

DOI: 10.54254/2755-2721/2025.GL25864

partitions MSB processing to digital units and LSB operations to analog modules through asymmetric zigzag allocation, creating an optimized digital-analog computing paradigm. Yuan et al. [4] proposed a 28-nm hybrid-domain FP-SRAM CIM macro by decomposing FP summation into analog bitwise multiplication and digital multi-bit shift-accumulation. Yue et al. [5] proposed a CV-CIM architecture combining pixel-CIM (digital XOR) and vector-CIM (current-based XOR) cores. This hybrid XOR-CIM design optimizes the digital/analog trade-off, enhancing cost-volume construction efficiency.

2.1.2. Hybrid computing with time domain

Time-domain CIM architectures leverage pulse-based temporal encoding for parallel accumulation but require extended latency for precision. Figure 2 shows this through its hybrid computing implementation.

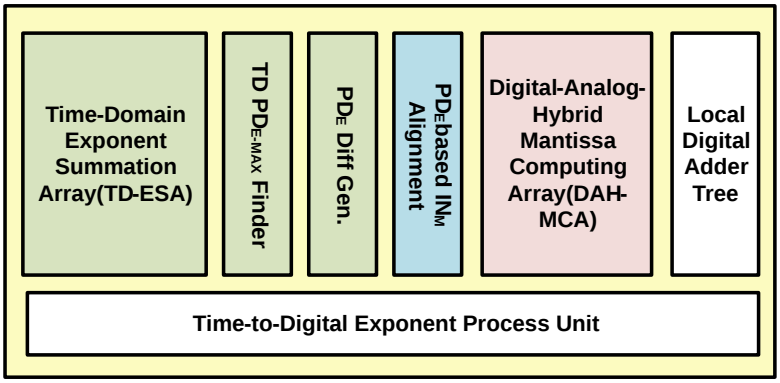

Figure 2. Time-domain exponent processing with hybrid computing

Wu et al. [1] proposed a hybrid-domain FP-CIM macro that processes exponents in time domain and mantissa in analog/digital domain. The 28nm prototype achieves 16.2-70.2 TFLOPS/W for FP16/FP32 edge AI, validating time-domain computing's energy-accuracy advantages. Jhang et al. [6] demonstrated a 22nm hybrid time-digital SRAM CIM accelerator employing two-stage digital computation for high-order partial products and tunable delay chains for low-order terms, achieving 10.03-237.99 TOPS/W at <0.3% accuracy loss. Lou et al. [7] proposed an all-digital time-domain CIM macro using standard-cell-based temporal encoding, achieving full EDA flow compatibility without analog components for enhanced scalability and practicality.

2.2. CIM storage circuit

CIM performance is heavily influenced by memory type. As single-memory architectures cannot optimize density, efficiency, and flexibility simultaneously, hybrid solutions have emerged to balance these limitations.

2.2.1. Hybrid SRAM/RRAM CIM

Hybrid SRAM/RRAM CIM architectures synergistically combine SRAM's fast dynamic computation with RRAM's dense, non-volatile storage, enabling concurrent optimization of energy efficiency, accuracy, and parallelism, as shown in Figure 3

DOI: 10.54254/2755-2721/2025.GL25864

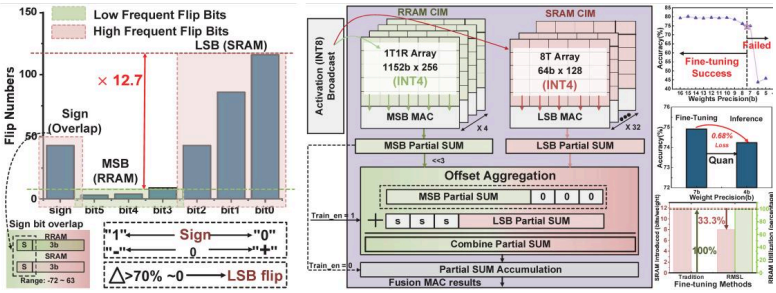

Figure 3. RRAM-MSB/SRAM-LSB collaborative CIM macros [8]

Chen et al. [9] developed a hybrid SRAM/RRAM CIM using peripheral circuits, combining SRAM reliability with RRAM density through bit-significance-aware weight allocation for energy-efficient parallel MAC operations. Yin et al. [10] implemented a 28nm hybrid RRAM/SRAM CIM accelerator with hierarchical bit-storage (RRAM: MSBs; SRAM: LSBs), achieving 75.49%/95.87% accuracy (ResNet18/ViT-Base), 183× RRAM update reduction, and 76.25 TOPS/W via bit-mask writing and sparse updates.

2.2.2. Hybrid ROM/SRAM CIM

The hybrid ROM/SRAM CIM architecture improves computing efficiency by leveraging complementary memory characteristics and employing separate static-dynamic data management. For edge AI applications, ROM stores fixed parameters, taking advantage of its high density, while SRAM manages dynamic weights and activations through its fast read/write capability to support real-time processing. Figure 4 shows this through a comparison of SRAM-CiM, YOLoC, and the proposed Hidden-ROM architecture.

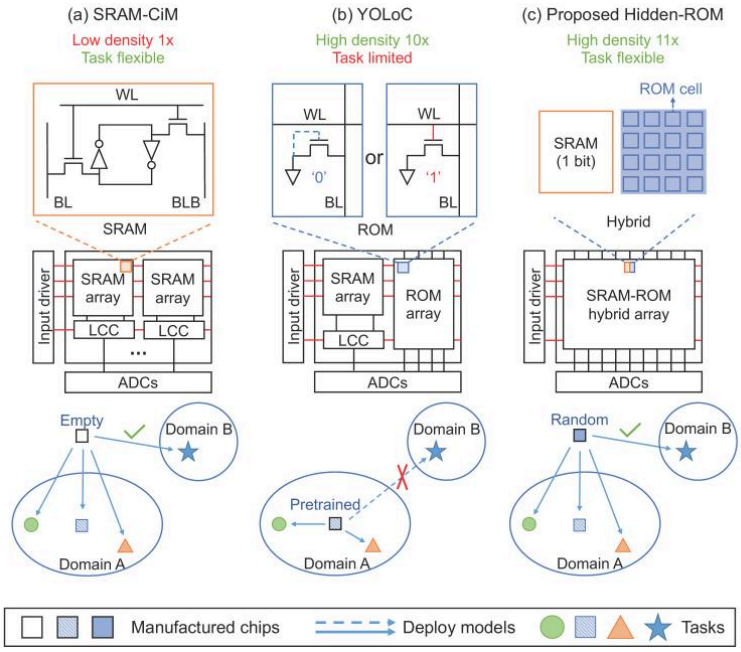

Figure 4. Comparison between (a) flexible but low-density SRAM-CiM, (b) dense but inflexible YOLoC, and (c) proposed dense and flexible Hidden-ROM [11]

DOI: 10.54254/2755-2721/2025.GL25864

Chen et al. [12] presented ReBranch, a hybrid ROM/SRAM CIM architecture where 90% of 8b fixed weights are stored in ROM, while less than 10% of 8b tunable weights are stored in SRAM using channel compression. ROM performs the primary computations during inference, and SRAM enables residual updates; only SRAM weights are trained.

Chen et al. [11] introduced a Hidden-ROM architecture based on HNN training, with random weights stored in ROM CIM and the learned topology in SRAM. During training, the HNN selects sub-networks from the ROM-stored random weights, and the final topology is mapped to SRAM. Yu et al. [13] developed an all-digital ROM-CIM achieving 27.9 Mb/mm² through weight sparsity compression. It performs 3×3 convolutions in ROM macros while offloading point-wise operations to SRAM CIM, resulting in a 94.4% area reduction.

2.2.3. Hybrid SRAM/MRAM CIM

The hybrid SRAM/MRAM CIM architecture leverages MRAM's high density for static weight storage and SRAM's low-latency write access for dynamic data, addressing the high write energy of MRAM and the volatility of SRAM.

Zhang et al. [14] proposed the HD-CIM architecture, a hybrid SRAM/MRAM CIM design that reduces weight-loading energy using MRAM's non-volatility and pipelined SRAM-CIM weight transfers. The design includes a dual-word-line MRAM array, a parallel interface with joint address decoding, and cross-coupled sense amplifiers to accelerate access, enhance data transfer, and improve read performance.

Zhang et al. [15] also introduced a hybrid MRAM/SRAM CIM accelerator for on-device continual learning, combining MRAM's density (for fixed weights) with SRAM's speed (for adapter updates). By incorporating structured sparsity and pipelined near-memory computing, the design achieves an efficient balance of capacity, energy, and flexibility.

3. Micro-architecture-level CIM macros

CIM research has evolved from fixed-point operations to FP and mixed-precision support while leveraging model robustness through approximate/sparse computing for area/power efficiency. This section analyzes CIM macro designs via data formats, approximation, and sparsity.

3.1. Data format evolution

3.1.1. Pure FP architecture

FP representation enables high-precision computation through sign-exponent-mantissa encoding, though it incurs higher energy and memory overhead than fixed-point formats. Hsu et al. [16] addressed this challenge in FP-CIM using a 22nm ReRAM macro with kernel-level weight pre-alignment, storing normalized mantissas in ReRAM and processing shared exponents digitally, thereby supporting analog multiplication with FP32 output while maintaining energy efficiency. He et al. [17] proposed a fully analog FP-CIM macro that employs gain amplifier arrays for analog exponent alignment, segmented modules for MSB detection, and counter-shared time-domain ADCs, enabling throughput-optimized FP operations. Guo et al. [18] implemented a 28nm digital FP-CIM macro based on Mitchell-approximated multiplication (MAM) for parallel accumulation, incorporating area-efficient 10T adders and sparse-aware power gating for energy savings. Ma et al. [19] demonstrated a 22nm hybrid-domain FP-CIM design achieving 72.14 TFLOPS/W for

DOI: 10.54254/2755-2721/2025.GL25864

BF16/FP32 formats through time-domain exponent processing and mixed-signal mantissa computation with exponent-aligned precision.

3.1.2. Hybrid INT/FP architecture

The INT-FP dual-mode CIM dynamically switches between integer and FP computation to address AI's precision-efficiency trade-offs, enabling high-throughput inference and precision-sensitive processing.

Tu et al. [20] developed ReDCIM. This reconfigurable digital CIM processor unifies FP/integer computation via exponent-aligned fixed-point conversion and in-memory Booth multiplication, improving latency and energy efficiency. Yan et al. [21] proposed a sparse-digital CIM architecture that processes dense FP clusters in memory while offloading sparse computations to digital cores, leveraging FP exponents' long-tail distribution for efficiency. Figure 5 shows the circuit schematics of the comparator and subtractor used in SiTe CiM-II, which exemplify the critical components enabling such mixed-precision computations. Guo et al. [22] proposed a CIM macro based on dual-bit 6T SRAM and hybrid computing units, optimizing energy efficiency and precision for edge AI applications. The architecture combines high-bit full-precision multipliers with low-bit approximate units to enable efficient FP MAC operations, utilizing a shared FP format to reduce exponent alignment overhead while incorporating reconfigurable units for seamless INT/FP switching.

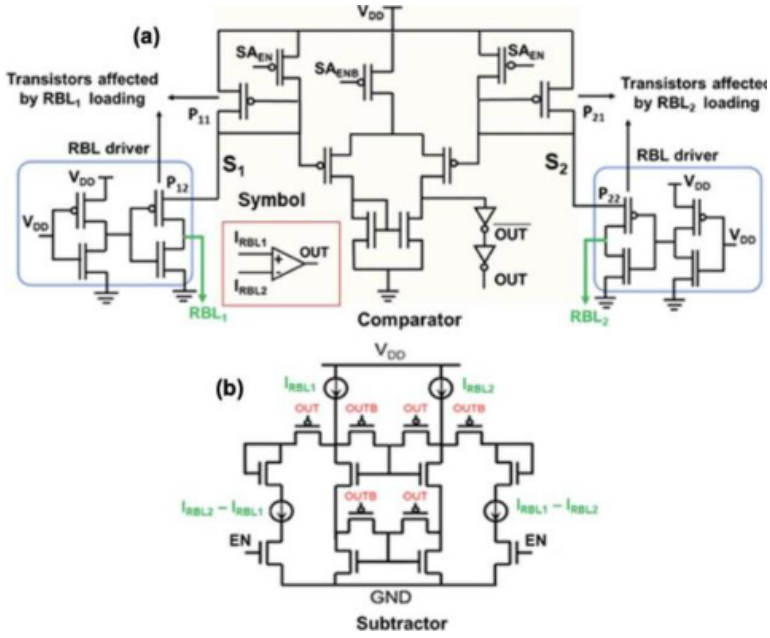

Figure 5. The circuit schematics of the (a) comparator and (b) subtractor used in SiTe CiM II [23]

3.1.3. Other number systems

Novel number systems (Posit, MX) enable CIM architecture advances through dynamic data encoding, overcoming IEEE 754's rigidity to match better AI computing's variable precision needs and data distributions.

The Posit number format can adjusts exponent/mantissa bit-widths based on value magnitude. Recent implementations demonstrate substantial improvements: Wang et al. [24] are 28nm DP-CIM macro leverages Posit's regime encoding to replace decoding circuits with shift-OR logic (40.3% energy reduction) while achieving 63% higher memory utilization through pre-computed weight

DOI: 10.54254/2755-2721/2025.GL25864

storage and 56.9% adder energy savings via OR-based accumulation. He et al. [25] 's AdaP-CIM optimizes this through adaptive encoding and speculative alignment (16.4% area/25.7% power reduction). Prabhu et al. [26] 's MINOTAUR edge SoC combines 8b dynamic Posit with RRAM/SRAM heterogeneity, enabling Transformer fine-tuning at 114mJ/sample through frozen RRAM weights and low-rank adaptation, marking Posit's first practical edge-training validation.

MX utilizes shared scaling factors (SS) for block-wise dynamic quantization in AI computing. Khwa et al. [27] successfully implemented this approach in a 16nm multi-mode (MX/INT/FP) CIM macro, featuring microscale-optimized dataflow, SS-FP circuit co-design for direct FP-MX conversion, and channel-adaptive data routing between memory arrays and ALU buffers to minimize data movement overhead.

3.2. Computational paradigms for optimization

3.2.1. Approximate calculation

Approximate computing in CIM architectures intentionally trades computational precision for enhanced hardware efficiency, making it ideal for error-tolerant neural network inference.

Recent work demonstrates significant advances in approximate CIM architectures. Diao et al. [28] achieved 102 TOPS/W in 28nm digital SRAM-CIM through multiply-less NN algorithms using L1-distance computation and dynamic approximation, maintaining <0.1% accuracy loss on ResNet models. Guo et al. [22] developed a dual-bit 6T SRAM-CIM with mixed-precision multiplication and dynamic exponent sharing, achieving accurate results with 1.6× area overhead. Further advancing FP efficiency, Guo et al. [18] implemented Mitchell-approximation-based FP-CIM with parallelized addition, reaching 65.15 TFLOPS/W@BF16 while reducing power by 12-27% through sparse computation.

3.2.2. Sparse computing

Sparse computing leverages neural network sparsity by exploiting zero-valued/low-contribution weights and activations to optimize computational resource utilization.

Yue et al. [29] developed a 65-nm CIM processor that combines block-level sparsity optimization, dynamic activation detection, and hybrid kernel/channel mapping to overcome sparse data processing limitations, achieving 158 TOPS/W (macro) and 35.8 TOPS/W (system) energy efficiency. Guo et al. [30] introduced a 28-nm TT@CIM processor with bit-level sparsity optimization, employing 1/2's complement hybrid encoding and TTD-CIM dataflow to enhance sparsity and reduce redundancy. Tu et al. [31] proposed a 28-nm MulTCIM accelerator for multimodal Transformers, utilizing attention-token-bit sparsity optimization with dynamic matrix reconstruction, token pruning, and adaptive CIM networks to improve computational efficiency. Um et al. [32] demonstrated an 88.9% read power reduction in their LOG-CIM processor through separated PMOS/NMOS word lines in 6T SRAM, with bidirectional pre-charge reuse achieving 60.7% lower CIM core power on ResNet-50@ImageNet. Guo et al. [33] proposed CIMFormer, a CIM accelerator for edge Transformers that integrates token-pruning-aware attention, sparse-parallel processing, and a systolic X|W-CIM array, achieving 15.71 TOPS/W at 28nm via joint token-bit optimization.

DOI: 10.54254/2755-2721/2025.GL25864

4. Emerging application on CIM

4.1. On-chip training

Existing CIM research for CNNs/Transformers has predominantly optimized inference efficiency for vision tasks. The increasing demand for adaptive AI systems now drives a paradigm shift toward on-chip training.

Recent advances address these challenges through novel CIM architectures. Yuan et al. [34] developed a 28nm digital transpose FP-SRAM CIM macro featuring cyclic weight mapping for forward/backward propagation reuse, supporting FP8/BF16 formats and dual-mode computation to optimize edge AI training/inference. Wang et al. [35] achieved simultaneous matrix-vector/element-wise multiplication in one clock cycle using a 14nm FinFET plastic CIM macro that integrates 5T logic flash with SRAM arrays. The design enables 4b/cell non-volatile weight adjustment via 5T-LF cells, implements synaptic plasticity through SRAM-stored Hebbian learning, and reduces ADC area overhead to 3.3% through array-level sharing, providing an efficient edge continual learning solution.

Mu et al. [8] 28nm RRAM/SRAM hybrid CIM architecture utilizes MSB/LSB hierarchical weight distribution and sparse updates to overcome RRAM endurance limitations, enabling accurate edge fine-tuning for both CNNs and Transformers. Diao et al. [36] 28nm CIM engine achieves single-cycle FP inference and efficient on-chip training through parallel minimum selectors and input-weight alignment, demonstrating rapid adaptation capability.

4.2. SLAM/NERF

SLAM and NeRF serve as core technologies for environmental perception and autonomous navigation in applications ranging from autonomous vehicles to AR/VR. Conventional CPU/GPU architectures struggle with the memory wall when processing SLAM's high-dynamic-range FP data.

Li et al. [37] solved this through a 28nm SLAM-CIM processor implementing dynamic-range-driven FP-MAC computation, which strategically skips insignificant mantissa operations based on exponent analysis, thereby reducing memory accesses and improving computational efficiency.

Guo et al. [38] overcome this through their 28nm NeuroPilot CIM macro, a 32×32 array employing dynamic logic steering and bidirectional search to achieve 69.4fJ/node energy efficiency at 0.9V while processing 3670M nodes/s, dramatically accelerating 3D reconstruction tasks. Jing et al. [39] developed a 22nm NeRF-Learner processor that unifies training and inference in a CIM architecture. The design incorporates BF16-precision bidirectional dataflow processing to reduce SRAM access energy by 71.8%, along with a hierarchical memory management unit (HUD-MMU) that achieves a 99% reduction in 3D scene feature storage through depth-aware strategies. Additionally, a spatiotemporal-aware ray scheduler dynamically optimizes computing resource allocation to critical scene regions. Jo et al. [40] developed a 28nm NeRPIM processor based on an 8T-SRAM CIM architecture, which incorporates bit-width-complementary quantization to reduce per-sample rendering energy by 59.9% and employs a 3D spatial tiling strategy to decrease global memory accesses by 98.6%.

4.3. Ising machine

Combinatorial optimization problems (COPs), being NP-hard, grow computationally intractable at scale but find widespread real-world applications.

DOI: 10.54254/2755-2721/2025.GL25864

Yue et al. [41] developed a scalable universal Ising machine using interaction-centric CIM with compressed graph-CSR sparse matrix storage in 40nm RRAM arrays. Their implementation achieves 100-100,000× speedup and superior energy efficiency versus CPU/GPU baselines. Building on this, Lo et al. [42] proposed a CIM-based ping-pong Ising architecture featuring bidirectional transposable memory arrays and bit-serial SAR ADCs, enabling efficient, fully connected graph annealing. Xie et al. [43] introduced Ising-CIM, which addresses combinatorial optimization through in-memory analog Hamiltonian computation using existing peripherals and cross-subarray King's graph mapping via phantom cells.

5. Conclusion

The rapid evolution of CIM technology has demonstrated its transformative potential in overcoming the von Neumann bottleneck, enabling energy-efficient, high-throughput computing for AI and big data applications. This review systematically analyzed CIM advancements across circuit design, architectural optimization, and emerging applications, highlighting key trends and future directions.

References

[1] P. -C. Wu et al., "A 22nm 832Kb Hybrid-Domain Floating-Point SRAM In-Memory-Compute Macro with 16.2-70.2TFLOPS/W for High-Accuracy AI-Edge Devices, " 2023 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2023.

[2] S. Jeong, J. Oh and D. Jeon, "A 28nm 157TOPS/W 446.9Kb/mm2 Compute-In-Memory SRAM Macro with Analog-Digital Hybrid Computing for Deep Neural Network Inference, " 2024 IEEE Custom Integrated Circuits Conference (CICC), Denver, CO, USA, 2024.

[3] A. Guo et al., "34.3 A 22nm 64kb Lightning-Like Hybrid Computing-in-Memory Macro with a Compressed Adder Tree and Analog-Storage Quantizers for Transformer and CNNs, " 2024 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2024.

[4] Y. Yuan et al., "34.6 A 28nm 72.12TFLOPS/W Hybrid-Domain Outer-Product Based Floating-Point SRAM Computing-in-Memory Macro with Logarithm Bit-Width Residual ADC, " 2024 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2024.

[5] s. Yue et al., "CV-CIM: A Hybrid Domain Xor-Derived Similarity-Aware Computation-in-Memory Supporting Cost-Volume Construction, " in IEEE Journal of Solid-State Circuits, vol. 60, no. 2.

[6] C. -J. Jhang et al., "A 22 nm 10.03-237.99 TOPS/W Time-Digital-Hybrid SRAM Compute-in-Memory AI Accelerator for GNN Edge Device Applications, " in IEEE Transactions on Circuits and Systems for Artificial Intelligence, vol. 1, no. 1.

[7] J. Lou, F. Freye, C. Lanius and T. Gemmeke, "An Energy Efficient All-Digital Time-Domain Compute-in-Memory Macro Optimized for Binary Neural Networks, " in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 71, no. 1.

[8] C. Mu et al., "A 28nm 76.25TOPS/W RRAM/SRAM-Collaborative CIM Fine-Tuning Accelerator with RRAM-Endurance/Latency-Aware Weight Allocation for CNN and Transformer, " 2024 IEEE Asian Solid-State Circuits Conference (A-SSCC), Hiroshima, Japan, 2024, pp.

[9] Y. -G. Chen, Z. -W. Liu and Y. -J. Tsai, "Exploring a Hybrid SRAM-RRAM Computing-In-Memory Architecture for DNNs Model Inference, " 2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Knoxville, TN, USA, 2024.

[10] G. Yin et al., "A 28nm 8928Kb/mm2-Weight-Density Hybrid SRAM/ROM Compute-in-Memory Architecture Reducing > 95% Weight Loading from DRAM, " 2024 IEEE Custom Integrated Circuits Conference (CICC), Denver, CO, USA, 2024.

[11] Y. Chen et al., "Hidden-ROM: A Compute-in-ROM Architecture to Deploy Large-Scale Neural Networks on Chip with Flexible and Scalable Post-Fabrication Task Transfer Capability, " 2022 IEEE/ACM International Conference On Computer Aided Design (ICCAD), San Diego, CA, USA, 2022.

[12] Yiming Chen et al., "YOLoC: deploy large-scale neural network by ROM-based computing-in-memory using residual branch on a chip, " In Proceedings of the 59th ACM/IEEE Design Automation Conference (DAC '22).

DOI: 10.54254/2755-2721/2025.GL25864

[13] T. Yu et al., "DSC-ROM: A Fully Digital Sparsity-Compressed Compute-in-ROM Architecture for on-Chip Deployment of Large-Scale DNNs, " 2025 Design, Automation & Test in Europe Conference (DATE), Lyon, France, 2025.
[14] H. Zhang et al., "HD-CIM: Hybrid-Device Computing-In-Memory Structure Based on MRAM and SRAM to Reduce Weight Loading Energy of Neural Networks, " in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 11.
[15] Fan Zhang et al., "Efficient Memory Integration: MRAM-SRAM Hybrid Accelerator for Sparse On-Device Learning, ". In Proceedings of the 61st ACM/IEEE Design Automation Conference (DAC '24).
[16] H. -H. Hsu et al., "A 22 nm Floating-Point ReRAM Compute-in-Memory Macro Using Residue-Shared ADC for AI Edge Device, " in IEEE Journal of Solid-State Circuits, vol. 60, no. 1.
[17] P. He et al., "A 28nm 314.6TLFOPS/W Reconfigurable Floating-Point Analog Compute-In-Memory Macro with Exponent Approximation and Two-Stage Sharing TD-ADC, " 2024 IEEE Custom Integrated Circuits Conference (CICC), Denver, CO, USA, 2024.
[18] R. Guo et al., "A 28nm 4170-TFLOPS/W/b and 195-TFLOPS/mm2/b Multiply-Free Fully-Digital Floating-Point Compute-In-Memory Macro with Mitchell's Approximation, " 2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), Honolulu, HI, USA, 2024.
[19] Z. Ma et al., "GCFP-ACIM: A 40nm 4.74TFLOPS/W General Complex Float-Point Analog Compute-in-Memory with Adaptive Power-Saving for HDR Signal Processing Applications, " 2023 IEEE Asian Solid-State Circuits Conference (A-SSCC), Haikou, China, 2023.
[20] F. Tu et al., "ReDCIM: Reconfigurable Digital Computing- In -Memory Processor With Unified FP/INT Pipeline for Cloud AI Acceleration, " in IEEE Journal of Solid-State Circuits, vol. 58, no. 1.
[21] S. Yan et al., "A 28-nm Floating-Point Computing-in-Memory Processor Using Intensive-CIM Sparse-Digital Architecture, " in IEEE Journal of Solid-State Circuits, vol. 59, no. 8.
[22] A. Guo et al., "A 28nm 64-kb 31.6-TFLOPS/W Digital-Domain Floating-Point-Computing-Unit and Double-Bit 6T-SRAM Computing-in-Memory Macro for Floating-Point CNNs, " 2023 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2023.
[23] Thakuria, Niharika, et al. "SiTe CiM: Signed Ternary Computing-in-Memory for Ultra-Low Precision Deep Neural Networks." arXiv preprint arXiv: 2408.13617 (2024).
[24] Y. Wang et al., "34.1 A 28nm 83.23TFLOPS/W POSIT-Based Compute-in-Memory Macro for High-Accuracy AI Applications, " 2024 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2024.
[25] J. He, F. Tu, K. -T. Cheng and C. -Y. Tsui, "AdaP-CIM: Compute-in-Memory Based Neural Network Accelerator Using Adaptive Posit, " 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE), Valencia, Spain, 2024.
[26] K. Prabhu et al., "MINOTAUR: A Posit-Based 0.42–0.50-TOPS/W Edge Transformer Inference and Training Accelerator, " in IEEE Journal of Solid-State Circuits, vol. 60, no. 4.
[27] W. -S. Khwa et al., "14.2 A 16nm 216kb, 188.4TOPS/W and 133.5TFLOPS/W Microscaling Multi-Mode Gain-Cell CIM Macro Edge-AI Devices, " 2025 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2025.
[28] H. Diao et al., "A Multiply-Less Approximate SRAM Compute-In-Memory Macro for Neural-Network Inference, " in IEEE Journal of Solid-State Circuits, vol. 60, no. 2.
[29] 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, " in IEEE Journal of Solid-State Circuits, vol. 57, no. 8.
[30] R. Guo et al., "TT@CIM: A Tensor-Train In-Memory-Computing Processor Using Bit-Level-Sparsity Optimization and Variable Precision Quantization, " in IEEE Journal of Solid-State Circuits, vol. 58, no. 3.
[31] F. Tu et al., "MulTCIM: Digital Computing-in-Memory-Based Multimodal Transformer Accelerator With Attention-Token-Bit Hybrid Sparsity, " in IEEE Journal of Solid-State Circuits, vol. 59, no. 1.
[32] S. Um, S. Kim, S. Hong, S. Kim and H. -J. Yoo, "LOG-CIM: An Energy-Efficient Logarithmic Quantization Computing-In-Memory Processor With Exponential Parallel Data Mapping and Zero-Aware 6T Dual-WL Cell, " in IEEE Journal of Solid-State Circuits, vol. 59, no. 10.
[33] R. Guo et al., "A Systolic CIM-Array-Based Transformer Accelerator With Token-Pruning-Aware Attention Reformulating and Principal Possibility Gathering, " in IEEE Journal of Solid-State Circuits, vol. 59, no. 10.
[34] Y. Yuan et al., "14.5 A 28nm 192.3TFLOPS/W Accurate/Approximate Dual-Mode-Transpose Digital 6T-SRAM CIM Macro for Floating-Point Edge Training and Inference, " 2025 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2025.
[35] L. Wang et al., "34.9 A Flash-SRAM-ADC-Fused Plastic Computing-in-Memory Macro for Learning in Neural Networks in a Standard 14nm FinFET Process, " 2024 IEEE International Solid-State Circuits Conference (ISSCC),

DOI: 10.54254/2755-2721/2025.GL25864

San Francisco, CA, USA, 2024.
[36] H. Diao et al., "A 28nm 128TFLOPS/W Computing-In-Memory Engine Supporting One-Shot Floating-Point NN Inference and On-Device Fine-Tuning for Edge AI, " 2024 IEEE Custom Integrated Circuits Conference (CICC), Denver, CO, USA, 2024.
[37] M. Li et al., "SLAM-CIM: A Visual SLAM Backend Processor With Dynamic-Range-Driven-Skipping Linear-Solving FP-CIM Macros, " in IEEE Journal of Solid-State Circuits, vol. 59, no. 11.
[38] A. Guo et al., "14.7 NeuroPilot: A 28nm, 69.4fJ/node and 0.22ns/node, 32×32 Mimetic-Path-Searching CIM-Macro with Dynamic-Logic Pilot PE and Dual-Direction Searching, " 2025 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2025.
[39] Y. Jing et al., "NeRF-Learner: A 2.79mJ/Frame NeRF-SLAM Processor with Unified Inference/Training Compute-in-Memory for Large-Scale Neural Rendering, " 2024 IEEE European Solid-State Electronics Research Conference (ESSERC), Bruges, Belgium, 2024.
[40] W. Jo et al., "NeRPIM: A 4.2 mJ/frame Neural Rendering Processing-in-memory Processor with Space Encoding Block-wise Mapping for Mobile Devices, " 2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), Kyoto, Japan, 2023.
[41] Yue, W., Zhang, T., Jing, Z. et al., "A scalable universal Ising machine based on interaction-centric storage and compute-in-memory, " Nat Electron 7, 904–913 (2024).
[42] C. -P. Lo, S. Oruganti, Y. Wang, M. Yang, S. Xie and J. P. Kulkarni, "Data Movement-Aware, Ping-Pong Ising Machine Supporting Full Connectivity and Variable Bitwidths, " 2024 IEEE European Solid-State Electronics Research Conference (ESSERC), Bruges, Belgium, 2024.
[43] S. Xie, S. R. S. Raman, C. Ni, M. Wang, M. Yang and J. P. Kulkarni, "Ising-CIM: A Reconfigurable and Scalable Compute Within Memory Analog Ising Accelerator for Solving Combinatorial Optimization Problems, " in IEEE Journal of Solid-State Circuits, vol. 57, no. 11.