PipeDCIM: 28nm 115.11TOPS/mm^2×TOPS/W@1.24GHz Pipeline Digital CIM Macro with an Auto-Design Tool for Diverse High-Performance AI Scenarios

Jia Chen[1,2], Tin-Chak Pang[2], Yat-Fong Yung[2], Yi Deng[2], Anqi Yin[2], Xiao Huo[2], Luhong Liang[2],
Zhongrui Wang[2,3], Chi-Ying Tsui[1,2], Kwang-Ting Cheng[1,2], Fengbin Tu[1,2*]
[1]The Hong Kong University of Science and Technology; [2]AI Chip Center for Emerging Smart Systems, Hong Kong, China;
[3]Southern University of Science and Technology, Shenzhen, China; *Corresponding Author (fengbintu@ust.hk)

***Abstract*—Large-scale AI computing requires balancing area and energy efficiency at the high-performance point. Traditional CIM designs often prioritize energy efficiency at the expense of frequency, limiting their applicability to high-performance scenarios. In this work, we propose PipeDCIM, a pipeline digital computing-in-memory macro targeting the FoM of TOPS/mm^2×TOPS/W. The contributions include: 1) TSPC-FF pipeline register with an 11T dynamic structure for area efficiency improvement. 2) Slack-power tuning strategy on non-critical pipeline stages for energy efficiency enhancement. 3) Auto-design tool for optimal pipeline architecture exploration and automating the proposed techniques. Two PipeDCIM macros were auto-generated by the tool and fabricated in 28nm. The best FoM reaches 115.11@1.24GHz, achieving 2.96~21.44× improvement over the state-of-the-art CIM macros.**

Keywords—Digital Computing-in-Memory Macro, Pipeline Architecture, TSPC-FF, Slack-Power Tuning

I. Introduction

Prior computing-in-memory (CIM) research usually focuses on optimizing energy efficiency but with sacrifice in frequency (<300MHz or even <100MHz [1-5]), which is reasonable for low-power AI scenarios. However, the rapid evolution of large-scale AI models [6, 7] raises a strong demand for CIM solutions with balanced area efficiency (TOPS/mm^2) and energy efficiency (TOPS/W) at the high-performance point. As shown in Fig. 1, TSMC has applied 2-stage and 3-stage pipeline architectures to digital CIM (DCIM) macros by segmenting in-memory combinational logic, raising the frequency to > 1GHz [8, 9]. To further optimize **the FoM of TOPS/mm^2×TOPS/W** under diverse scenarios, designing pipeline DCIM macros faces three challenges (Fig. 2): 1) **Area Efficiency**: Introducing pipeline registers into CIM macros causes substantial area overhead, which increases with stage count. 2) **Energy Efficiency**: Non-critical pipeline stages often have unused latency slacks that can be leveraged to save power without degrading performance. 3) **Design Complexity**: Pipeline stage count and positions affect performance-power-area (PPA) tradeoffs. Determining the optimal pipeline DCIM architecture for one scenario needs to explore a large design space. Previous CIM designs heavily rely on manual efforts, which is time-consuming and inefficient for exploration.

To address these challenges, we propose **PipeDCIM**, a pipeline digital CIM (PipeDCIM) macro targeting the FoM with an auto-design tool: 1) The pipeline register is designed with 11T true single-phase clock flip-flop (**TSPC-FF**), obtaining higher area density with elimination of retention issues by utilizing the full pipelining manner. 2) PipeDCIM realizes **slack-power tuning** on non-critical stages with sensitivity-aware threshold voltage (Vt) assignment. Under the latency constraint from the critical stage, we design latency-insensitive transistors with high Vt to reduce the non-critical stage slacks and save leakage power. 3) We develop an **auto-design tool** based on the scalable PipeDCIM macro template to explore various pipeline stage count and positions. The former two techniques are integrated for automated circuit generation.

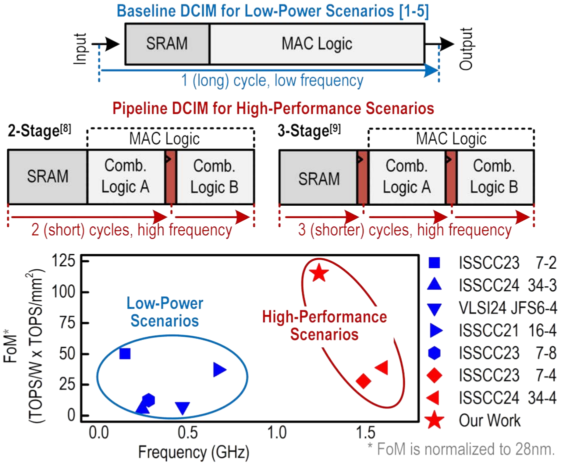


Fig. 1. Illustration of baseline DCIM (w/o pipeline) for low-power scenarios and pipeline DCIM for high-performance scenarios.

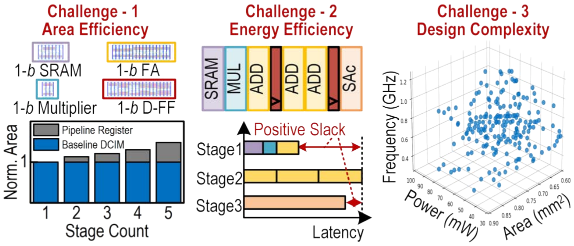


Fig. 2. Pipeline DCIM macro design challenges with area-energy efficiency FoM as the target.

II. PipeDCIM Overall Architecture

Fig. 3 shows the PipeDCIM macro template used in the design tool, which provides a scalable architecture for diverse scenario requirements. Generally, the macro is divided into different functional modules, including SRAM, bitwise-multiplier (MUL), adder tree (AddT), and shift-accumulator (SAc) parts with peripherals for input feeding and output fusion. Pipeline stages can be inserted between different parts and inner levels of AddT. The

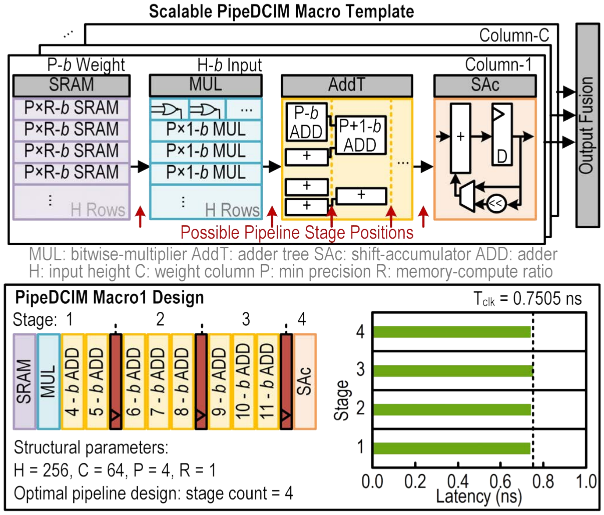


Fig. 3. Scalable PipeDCIM macro template used in the design tool and an example of auto-generated 256×256b 4-stage PipeDCIM Macro1.

function and structure are defined by parameters including input height (H), weight column (C), minimum data precision (P), and memory-compute ratio (R). Each part is constructed by the DCIM modules pre-implemented in the library (i.e. P×R-b SRAM, P×1-b MUL, P-b ADD), where these modules are tunable by only adjusting the basic functional cells with minimum manual efforts. The bottom of Fig. 2 shows an example of an auto-generated 256×256b 4-stage PipeDCIM with a simulated critical path delay of 0.7505ns. Fabrication in 28nm, its measured frequency is 1.24GHz (Macro1 in Fig. 9). We will use this macro as the running case for demonstrating the proposed techniques in the following circuit implementation section.

III. Circuit Implmentation

A. TSPC-FF Pipeline Register for Higher Area Efficiency

Compared to the standard D-type flip-flop (D-FF), TSPC-FF (Fig. 4) is an 11T dynamic circuit without reset logic, reducing 17 transistors, saving area by 2.63× and power consumption by 1.44×. TSPC-FF benefits from the precharge mechanism of dynamic circuits, obtaining a short Clk-to-Q propagation delay (<170ps) under a wide voltage range for stable high-frequency operations. During CIM initialization, the inputs automatically reset all pipeline-stage registers, eliminating the need for reset logic. A critical challenge with dynamic circuits is data retention, as charge leakage can corrupt stored values. However, the PipeDCIM architecture inherently mitigates this issue through its high-frequency operation (clock period <1ns), which refreshes pipeline stages before retention failures occur. Experimental measurements confirm that the TSPC-FF retains data integrity for up to 227.8ns - far exceeding the sub-ns refresh requirement.

As shown in Fig. 5, the TSPC-FF approach saves pipeline register area by 2.24×, decreasing corresponding ratio to 8.69% of the whole macro. Besides area saving, TSPC-FF also reduces power by 1.20×. With proposed TSPC-FF optimization, Macro1 achieves 1.22× area efficiency and 1.61× FoM over the D-FF-based baseline, and implements up to 6.99× higher FoM than the non-pipeline baseline.

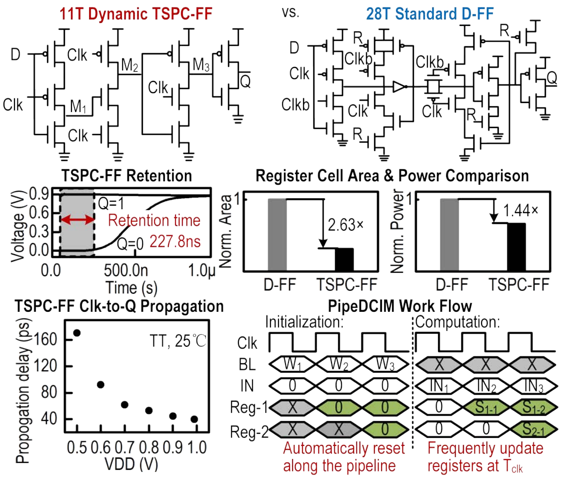


Fig. 4. Specification comparison between TSPC-FF and D-FF. PipeDCIM's work flow enables automatic reset and avoids the retention issue.

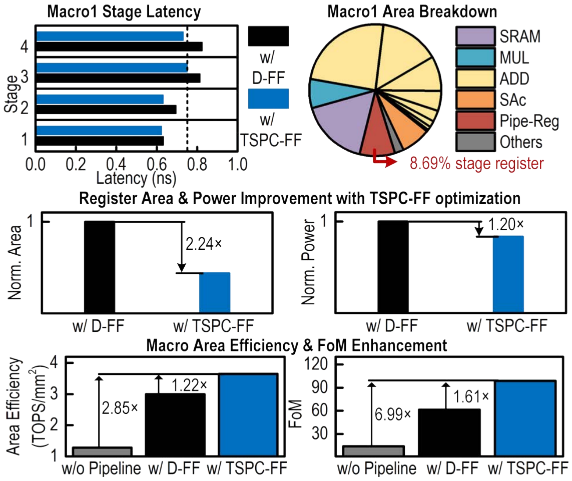


Fig. 5. Area efficiency and FoM enhancement with TSPC-FF as pipeline registers for PipeDCIM Macro1.

B. Slack-Power Tuning for Higher Energy Efficiency

Pipelining introduces latency slacks in non-critical stages, which are usually underutilized. Since only the critical-path stage determines the operating frequency, we propose sensitivity-aware Vt assignment to tune the slacks of non-critical stages for power reduction. By analyzing the signal propagation of PipeDCIM functional cells (Fig. 6), we identify the latency-insensitive transistors (in red), including the writing & storage MOS of 8T-SRAM, parallel-connect pull-down MOS of NOR gates, and bit-independent sum MOS of full adders (FA). Since these transistors aren't in the cell-level latency-critical paths, they are strategically replaced with high-Vt variants, which exhibit reduced subthreshold leakage currents at the cost of slightly increased switching delays.

Fig. 7 presents hierarchical tuning process with three Vt levels (S: standard Vt, H: high Vt, uH: ultra-high Vt) and final macro-level tuning results for Macro1. For example, in Macro1, Stage3 is the critical path. Stage4 has a quite short slack, so tuning only brings 0.90% lower power. Stage1 and Stage2 with longer slack, however,

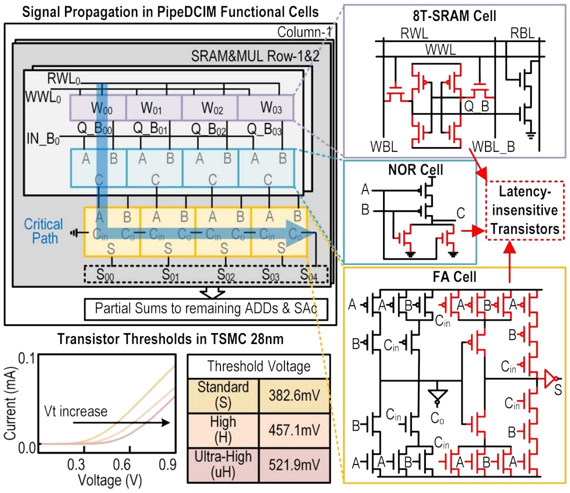
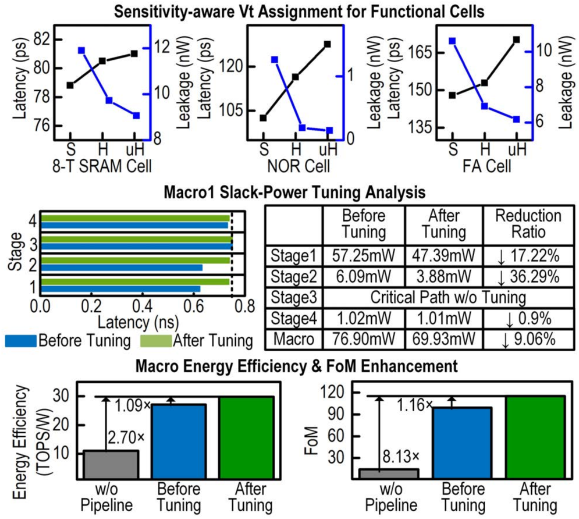


Fig. 6. Signal propagation analysis and latency-insensitive transistor identification (in red) for the functional cells of a PipeDCIM macro.



	Before Tuning	After Tuning	Reduction Ratio
Stage1	57.25mW	47.39mW	↓ 17.22%
Stage2	6.09mW	3.88mW	↓ 36.29%
Stage3	Critical Path w/o Tuning		
Stage4	1.02mW	1.01mW	↓ 0.9%
Macro	76.90mW	69.93mW	↓ 9.06%

Fig. 7. Energy efficiency and FoM enhancement with sensitivity-aware Vt assignment for slack-power tuning on PipeDCIM Macro1.

obtain 1.18× and 1.17× latency increase without exceeding Stage3, achieving 17.22% and 36.29% lower power consumption. In the pipeline architecture, the timing of non-critical stages doesn't affect the critical path, so the operating frequency is maintained. Overall, slack-power tuning reduces Macro1's power by 9.06%, leading to 8.13× higher FoM than the non-pipeline baseline.

C. PipeDCIM Auto-Design Tool for Agile Development

Designing optimal pipeline architectures requires navigating a vast design space spanning stage counts, pipeline positions, and Vt assignments. Manual exploration is impractical due to combinatorial complexity, necessitating an automated solution. Fig. 8 shows the PipeDCIM-supported auto-design tool, comprising three steps: DCIM library setup, DCIM pipeline exploration, and slack-power tuning. [STEP0] Develop a customized cell library including SRAM, FA, and TSPC-FF cells. Develop a module library with design files of all types of modules that comprise PipeDCIM to support diverse CIM configurations. [STEP1] Explore the design space with different stage count and positions based on the PipeDCIM template (in Fig. 3) and user-defined architecture specifications. Under user-defined PPA constraints, a Pareto boundary is formed by rapid PPA evaluation with the module library. The obtained highest-area efficiency point indicates the optimal pipeline architecture. [STEP2] Perform further slack-power tuning on the non-critical stages of this design point. By exploring different Vt assignments, each non-critical stage's latency is increased to the near-critical level, thus minimizing leakage power and maximizing FoM. The tool generates the layout of the highest-FoM macro based on the module library. Besides the presented features, this auto-design tool is friendly to extension for more advanced technology, macro architecture, and circuit techniques.

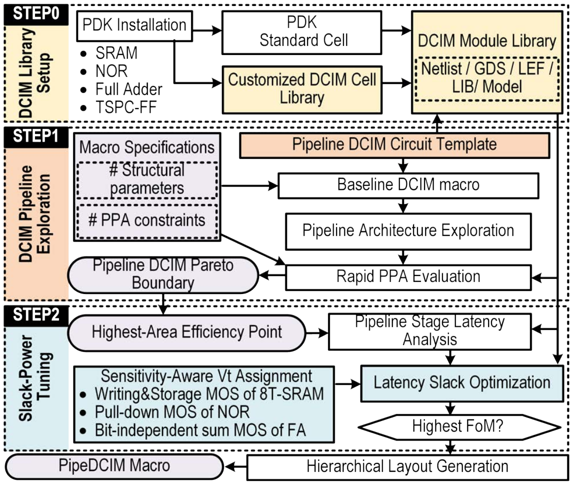

STEP0
DCIM Library Setup
PDK Installation
• SRAM
• NOR
• Full Adder
• TSPC-FF
PDK Standard Cell
Customized DCIM Cell Library
DCIM Module Library
Netlist / GDS / LEF / LIB/ Model
STEP1
DCIM Pipeline Exploration
Macro Specifications
# Structural parameters
# PPA constraints
Pipeline DCIM Circuit Template
Baseline DCIM macro
Pipeline Architecture Exploration
Pipeline DCIM Pareto Boundary
Rapid PPA Evaluation
STEP2
Slack-Power Tuning
Highest-Area Efficiency Point
Pipeline Stage Latency Analysis
Sensitivity-Aware Vt Assignment
• Writing&Storage MOS of 8T-SRAM
• Pull-down MOS of NOR
• Bit-independent sum MOS of FA
Latency Slack Optimization
Highest FoM?
PipeDCIM Macro
Hierarchical Layout Generation

Fig. 8. PipeDCIM auto-design tool, with DCIM library setup, DCIM pipeline exploration, and slack-power tuning as three steps.

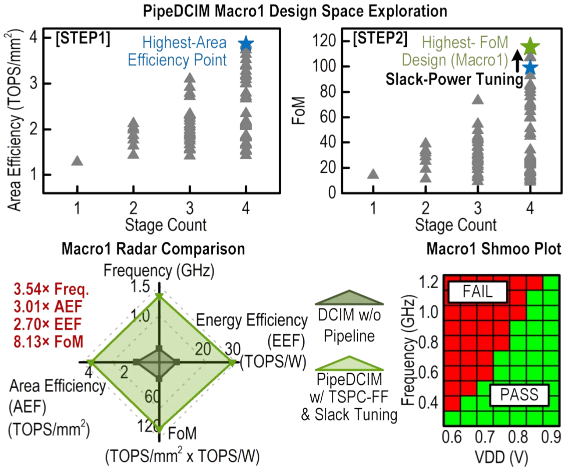
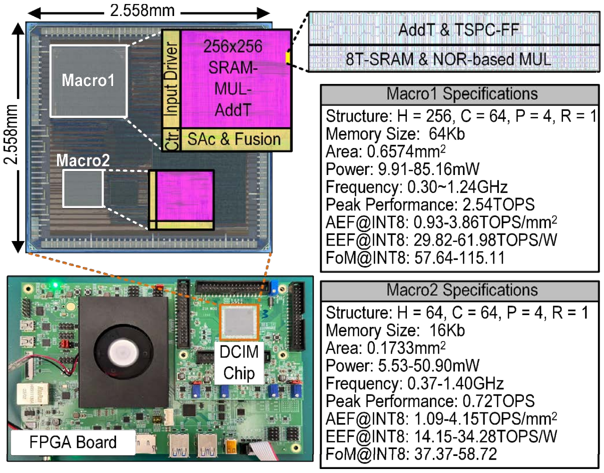


Fig. 9. PipeDCIM Macro1's design space exploration illustration, radar comparison with the baseline, and measured shmoo plot.

IV. Measurement Results

Fig. 9 presents PipeDCIM Macro1's design space exploration, radar comparison with the non-pipeline



Macro1 Specifications
Structure: H = 256, C = 64, P = 4, R = 1
Memory Size: 64Kb
Area: 0.6574mm^2
Power: 9.91-85.16mW
Frequency: 0.30~1.24GHz
Peak Performance: 2.54TOPS
AEF@INT8: 0.93-3.86TOPS/mm^2
EEF@INT8: 29.82-61.98TOPS/W
FoM@INT8: 57.64-115.11

Macro2 Specifications
Structure: H = 64, C = 64, P = 4, R = 1
Memory Size: 16Kb
Area: 0.1733mm^2
Power: 5.53-50.90mW
Frequency: 0.37-1.40GHz
Peak Performance: 0.72TOPS
AEF@INT8: 1.09-4.15TOPS/mm^2
EEF@INT8: 14.15-34.28TOPS/W
FoM@INT8: 37.37-58.72

Fig. 10. Chip die photo, FPGA-based test platform, and summary tables for two fabricated PipeDCIM macros (Macro1 and Macro2).

baseline, and shmoo plot. Under the specified parameters and FoM target, Macro1's circuit implementation details are determined in the tool's two-step optimization. Fabricated in 28nm CMOS, Macro1 work at 0.6~0.9V, 0.30~1.24GHz, with an FoM of 115.11@1.24GHz, INT4. In comparison with the non-pipeline baseline, Macro1 achieves 8.13× FoM with 3.54× frequency, 3.01× area efficiency, 2.70× energy efficiency. Another 64×256b 4-stage PipeDCIM macro (Macro2) is also fabricated to validate the auto-design tool's scalability. Fig. 10 depicts the die photo, test platform, and summary tables for the fabricated two PipeDCIM macros. Table I shows the comparison with state-of-the-art CIM macros. Our Macro1's pipeline architecture achieves 1.77~5.08× higher max frequency than the three non-pipelining CIM macros [3, 4, 9]. The frequency is even comparable to the TSMC DCIM macro with pipelining (1.60GHz) in more advanced 3nm nodes [8]. Overall, Macro1 achieves 2.96~21.44× FoM improvement, proving PipeDCIM's balance in area and energy efficiency to meet the requirement of high-performance AI scenarios.

V. Conclusion

This paper introduces the PipeDCIM macro which targets the FoM of TOPS/mm^2×TOPS/W with circuit-level innovations assisted by an auto-design tool. By integrating dynamic TSPC-FF registers, slack-aware power tuning, and an auto-design tool, PipeDCIM achieves up to 21.44× FoM improvement over published state-of-the-art CIM macros. The silicon-proven results indicate that the auto-design tool builds an agile development methodology for PipeDCIM macros and an ecosystem for CIM technology, thus catching up with the fast-evolving AI era.

Acknowledgment

This research was supported in part by NSFC Grant 62422407, RGC Grant 26204424, C6003-24Y, and partially conducted by ACCESS – AI Chip Center for Emerging Smart Systems, supported by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government.

Table I Comparison with the state-of-the-art CIM macros

	ISSCC21 16.4 [9]	VLSI24 JFS6-4 [4]	ISSCC24 34.3 [3]	ISSCC24 34.4 [8]	**This Work (Macro1)**
CIM Type	Digital	Digital	Hybrid	Digital	**Digital**
Arch. Opt.	×	×	×	√ Pipeline	**√ Pipeline**
Register Opt.	N/A	N/A	N/A	×	**√ TSPC-FF**
Slack Opt.	N/A	N/A	N/A	×	**√ Slack-power tuning**
Technology	22nm	28nm	22nm	3nm	**28nm**
Voltage (V)	0.52-0.92	0.56-0.9	0.6-0.9	0.4-1.0	**0.6-0.9**
Freq.(GHz)	0.06-0.7	0.05-0.4	0.096-0.244	0.3-1.6	**0.30-1.24**
Memory(Kb)	64	144	64	60.75	**64**
Precision	INT4-16	INT4/8	INT8	INT12	**INT4/8**
Area (mm²)	0.202	4.28	0.119	0.0157	**0.6574**
Area Eff.*1 (TOPS/mm²)	4.54	0.43	0.53	123.82*2	**3.86**
Energy Eff.*1 (TOPS/W)	15.81	16.73	20.7	27.36*4	**29.82*3**
FoM *5	34.78	7.19	5.37	38.88	**115.11**

*1: INT8 result@ high frequency. *2: The INT8 results are projected from the INT4 numbers reported in [8]. *3: Weight sparsity = 50%, input toggle = 25%. *4: Scaled from the 0.55V result in [8]. *5: Normalized to 28nm.

References

[1] Guo A, Xi C, Dong F, et al. A 28-nm 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). IEEE, 2023: 128-129.

[2] Guo A, Chen X, Dong F, 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). IEEE, 2024, 67: 570-572.

[3] Dai Z, Yan S, Cong Z, et al. A 41.7 TOPS/W@ INT8 computing-in-memory processor with Zig-Zag backbone-systolic CIM and block/self-gating CAM for NN/recommendation applications. 2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits). IEEE, 2024: 1-2..

[4] Chih Y D, Lee P H, Fujiwara H, et al. 16.4 An 89TOPS/W and 16.3 TOPS/mm 2 all-digital SRAM-based full-precision compute-in memory macro in 22nm for machine-learning edge applications. 2021 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2021, 64: 252-254.

[5] Chen P, Wu M, Zhao W, et al. 7.8 A 22nm delta-sigma computing-in-memory (Δ Σ CIM) SRAM macro with near-zero-mean outputs and LSB-first ADCs achieving 21.38 TOPS/W for 8b-MAC edge AI processing. 2023 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2023: 140-142..

[6] Tu F, Wu Z, Wang Y, et al. 16.1 MuITCIM: A 28nm 2.24uJ/Token Attention-Token-Bit Hybrid Sparse Digital CIM-Based Accelerator for Multimodal Transformers. 2023 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2023: 248-250.

[7] Kim S, Li Z, Um S, et al. 16.5 DynaPlasia: An eDRAM in-memory-computing-based reconfigurable spatial accelerator with triple-mode cell for dynamic resource switching. 2023 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2023: 256-258.

[8] Mori H, Zhao W C, Lee C E, et al. A 4nm 6163-TOPS/W/b 4790-TOPS/mm^2 SRAM Based Digital-Computing-in-Memory Macro Supporting Bit-Width Flexibility and Simultaneous MAC and Weight Update. 2023 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2023: 132-134.

[9] Fujiwara H, Mori H, Zhao W C, et al. 34.4 A 3nm, 32.5 TOPS/W, 55.0 TOPS/mm 2 and 3.78 Mb/mm 2 Fully-Digital Compute-in-Memory Macro Supporting INT12× INT12 with a Parallel-MAC Architecture and Foundry 6T-SRAM Bit Cell. 2024 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2024, 67: 572-574.