A 12nm 80.6TFLOPS/W BF16 Floating-Point Compute-in-Memory Macro with In-Memory Exponent Search and Shift-Free Mantissa Alignment for LLM Decoding

Siqi He[1], Jiapei Zheng[1], Kaijun Zhang[1], Zexing Chen[2], Yunzhengmao Wang[1], Qi Liu[1], Chixiao Chen[1], Haozhe Zhu[1]
[1]State Key Laboratory of Integrated Chips and Systems (SKLICS), Fudan University, China, [2]CiMicro.Ai, China
E-mail: zhuhz@fudan.edu.cn, cxchen@fudan.edu.cn

***Abstract*—This paper presents a 12nm floating-point compute-in-memory (FP-CIM) macro targeting large language model (LLM) decoding. To eliminate the latency and power overhead of explicit exponent comparison and mantissa alignment, we introduce: (1) a magnitude-search content-addressable memory (MS-CAM) enabling parallel exponent comparison, (2) an orthogonal-accessed SRAM (OA-SRAM) for pattern-indexed mantissa retrieval, and (3) a latch-based alignment-pattern (LAP) generator for shift-free mantissa alignment. A prototype chip fabricated in 12nm FinFET achieves a peak energy efficiency of 80.6 TFLOPS/W in BF16, demonstrating 1.2-1.7× higher energy efficiency than prior FP-CIM designs.**

***Index Terms*—Compute-in-memory, floating-point, implicit magnitude search, large language model.**

I. Introduction

CIM architectures have been widely explored to improve energy efficiency (EF) in AI workloads [1-7]. However, LLM inference is typically dominated by the decoding stage, where limited weight reuse reduces the effectiveness of weight-stationary dataflows and instead favors activation-stationary (AS) execution [8]. For floating-point (FP) LLM deployment, AS-based CIM introduces a fundamental challenge that dynamic activations cannot be pre-aligned offline like weights, requiring runtime alignment support. Prior FP-CIM alignment schemes rely on explicit compare-and-shift operations, which suffer from two key limitations. First, exponent alignment via parallel comparison or sequential search leads to an unfavorable latency-power trade-off (Fig. 1) [1,6]. Second, mantissa alignment using shift registers or barrel shifters introduces significant area and energy overhead (Fig. 2) [5,7].

To address these, we propose an implicitly-aligned FP-CIM architecture with three key innovations: (1) a magnitude-search content-addressable memory (MS-CAM) exponent array to accelerate exponent comparison while reducing power consumption; (2) an orthogonal-accessed SRAM (OA-SRAM) mantissa array to facilitate flexible pattern-indexed mantissa retrieval; and (3) a latch-based alignment-pattern (LAP) generator to implement low-cost implicit shifts via incremental thermometer encoding. To validate the proposed design, a 1 kb FP-CIM macro is fabricated in 12nm and achieves an EF of 80.6 TFLOPS/W using BF16, corresponding to a 1.2-1.7× improvement over state-of-the-art CIM designs [3-5].

Haozhe Zhu and Chixiao Chen are corresponding authors.

This work was supported by National Natural Science Foundation of China (NSFC) under grant No. 62488101 and No. 62495101.

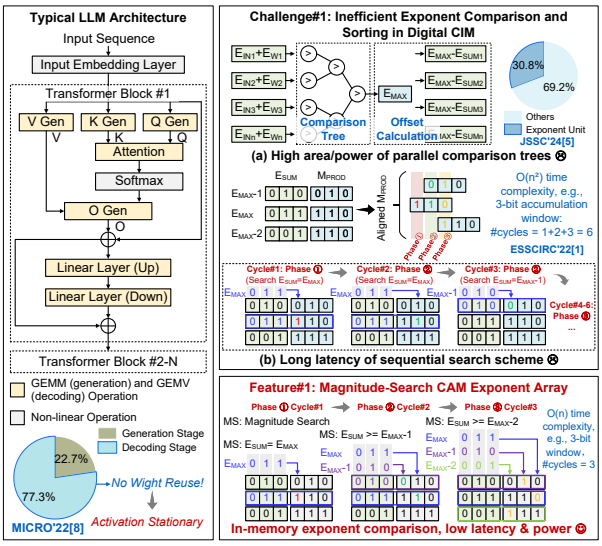


Fig. 1. Typical LLM structure, the primary challenge in designing FP-CIM for LLMs, and the proposed Feature 1 (MS-CAM).

II. Proposed FP-CIM Macro

A. Overall Architecture

Fig.3 shows the overall architecture and operation flow of the proposed FP-CIM macro. It consists of a 64×8b MS-CAM array for exponent storage and an interleaved 128×8b mantissa array formed by OA-SRAM cells and LAP cells. It also includes peripheral circuits for read, write, and search operations, along with a clock generation array, an E_{MAX} detector, a shift-and-accumulation module, and a quantizer.

The FP-CIM macro supports two operation modes. In the write mode, activation data are stored into the MS-CAM and OA-SRAM arrays through the write interface. During this process, the E_{MAX} detector continuously updates the maximum exponent by comparing incoming data with the current maximum. The computation mode consists of four iterative stages. In the first stage, the MS-CAM performs a range search over all stored exponents using the value provided by the E_{MAX} detector (initialized as E_{MAX}), and match lines (MLs) are asserted accordingly. In the second stage, the clock generation array produces clock pulses according to the asserted MLs and feeds them into the LAP array to generate alignment patterns. In the third stage, based on the alignment patterns, the OA-SRAM array reads out 64 aligned 1-bit

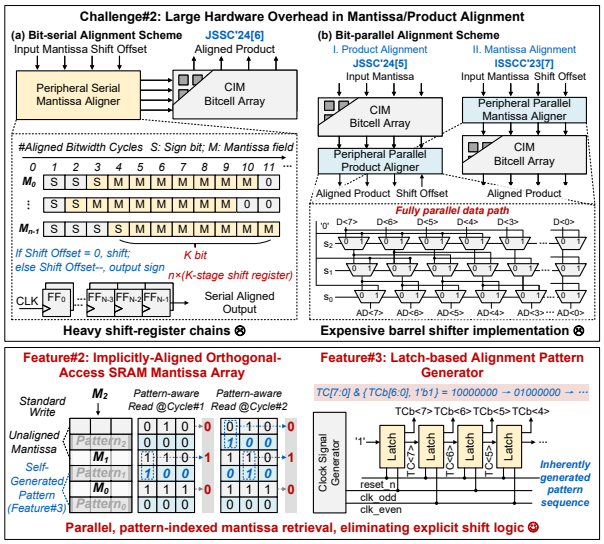


Fig. 2. The second challenge, along with the proposed Feature 2 (OA-SRAM) and Feature 3 (LAP).

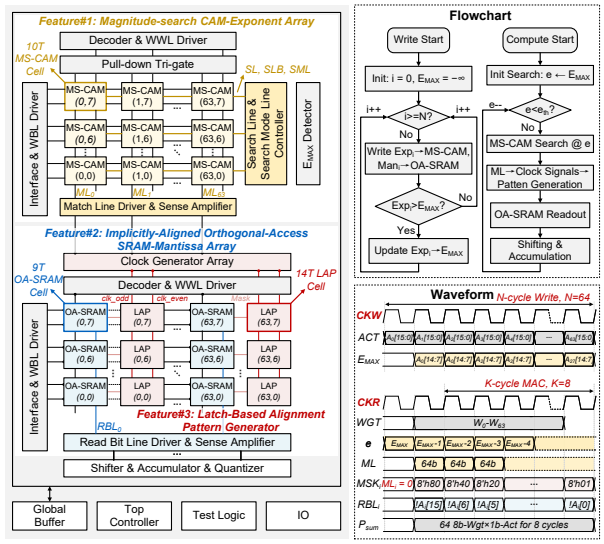


Fig. 3. Overall macro architecture.

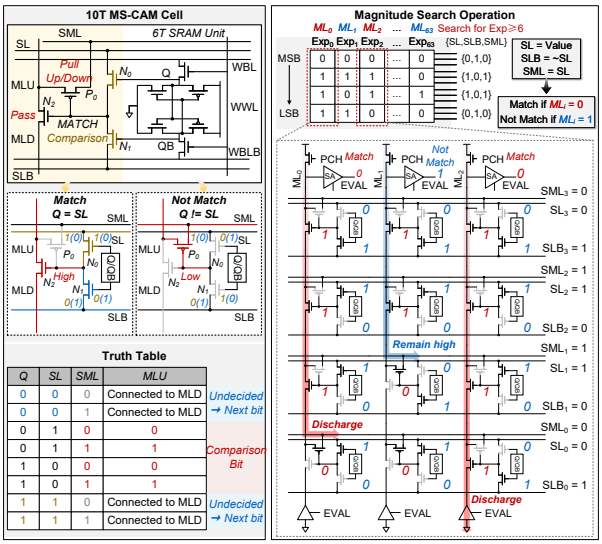


Fig. 4. Proposed MS-CAM cell design.

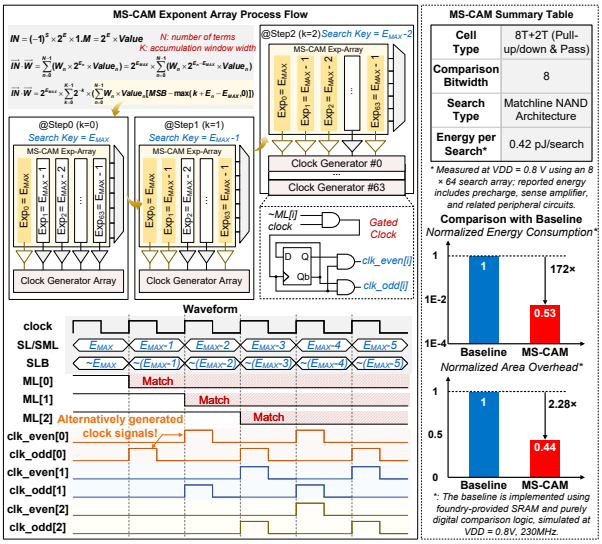


Fig. 5. Workflow of exponent processing using MS-CAM.

activation mantissas along the direction orthogonal to the write path. In the fourth stage, the 64 aligned 1-bit activations are multiplied with 64 corresponding weights to compute a partial sum. This sequence repeats for K cycles with a decremented search value, followed by shift-accumulate operations and BF16 quantization.

B. Magnitude-Search Content-Addressable Memory Exponent Array (MS-CAM)

Fig. 4 (left) details the proposed MS-CAM for parallel exponent comparison and E_{MAX} detection. Each cell consists of a 6T SRAM for data storage, two transistors ($\mathrm{N_0}$ and $\mathrm{N_1}$) for comparison, and two transistors ($\mathrm{N_2}$ and $\mathrm{P_0}$) for decision making. According to the truth table, when $\mathrm{SL} = \mathrm{Q}$, the MATCH node is high, $\mathrm{N_2}$ is turned on, and MLU is connected to MLD. When $\mathrm{SL} \neq \mathrm{Q}$ the MATCH node is low, $\mathrm{P_0}$ is turned on, and MLU is connected to SML. Fig. 4 (right) illustrates the multi-bit search operation. In the MS-CAM array, all bits of one exponent are mapped to a single column and arranged from the most significant bit (MSB) to the least significant bit (LSB). These cells are vertically connected to form a NAND-type match-line structure. During the precharge phase, PCH=1, SML remains in a high-impedance state, and ML is precharged to VDD. During the evaluation phase, PCH=0 and EVAL=1. To identify entries larger than a given Value, SL and SML are set to Value, while SLB is set to its complement. If Exp_i equals Value, as in Exp_2, all $\mathrm{N_2}$ devices in the column are turned on, discharging ML to GND through the enabled tri-state gates. If Exp_i is smaller than Value, as in Exp_1, there exists a bit position n such that $\mathrm{Exp}_i[n] = 0$ and $\mathrm{SL}[n] = \mathrm{SML}[n] = 1$. In this case, $\mathrm{P_0}$ is turned on, and ML is connected to SML, thus remaining at a high level. If Exp_i is larger than Value, as in Exp_0, there exists a bit position n such that $\mathrm{Exp}_i[n] = 1$ and $\mathrm{SL}[n] = \mathrm{SML}[n] = 0$.

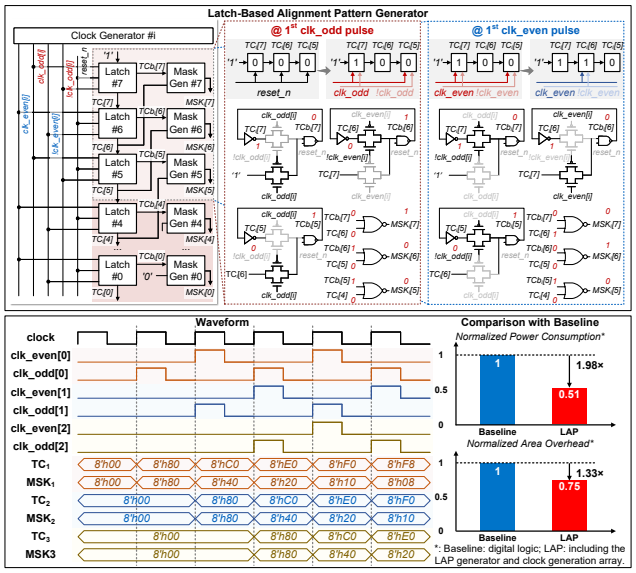


Fig. 6. Proposed LAP generation scheme and the related workflow.

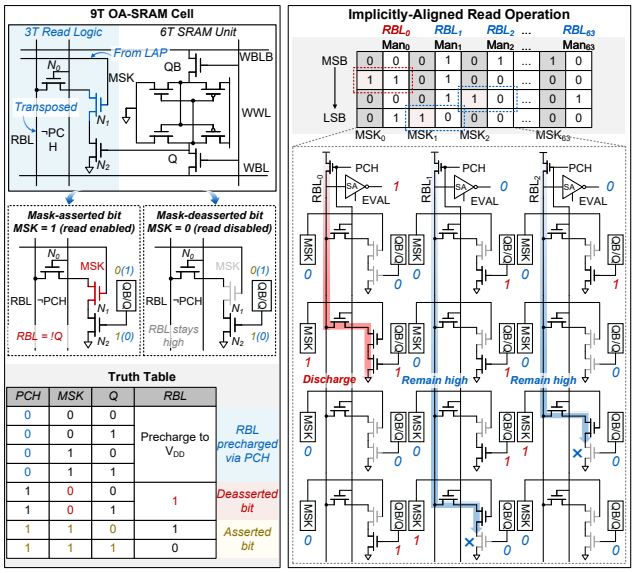


Fig. 7. Proposed OA-SRAM cell design.

In this case, P_0 is turned on, and ML is connected to SML and discharged to GND. After evaluation, the ML result is latched by a sense amplifier, where 0 indicates a match and 1 indicates a mismatch.

Fig. 5 shows the overall exponent processing. To support a K-size mantissa accumulation window, MS-CAM performs magnitude searches from E_{MAX} to $E_{\mathrm{MAX}} - (K-1)$, and the asserted MLs drive the clock generation array to produce alternating clock pulses. Compared to a baseline using standard SRAM and digital comparison logic, MS-CAM achieves a 172× reduction in power and a 2.28× reduction in area, enabling fast, parallel exponent comparison at minimal cost.

C. Latch-based Alignment-Pattern Generator (LAP)

Fig. 6 shows the proposed LAP scheme, which encodes alignment positions with a monotonic thermometer code (TC), rather than relying on explicit shift operations. As illustrated in Fig. 6 (right), a latch chain driven by alternating clk_{odd} and clk_{even} propagates a single injected '1' by one position per cycle to generate the TC. The TC 1→0 transition is detected by the NOR logic, therefore producing a one-hot mask serving as an index for mantissa access. Exploiting the single-transition property of the TC, the latch-based scheme eliminates DFF-based shift registers and wide shift networks, achieving equivalent functionality at only 51% of the baseline power and 25% of the baseline area.

D. Orthogonal-Accessed SRAM Mantissa Array (OA-SRAM)

Fig. 7 shows the proposed OA-SRAM, which cooperates with the LAP to embed mantissa alignment as an intrinsic memory-level operation. Each cell comprises a 6T SRAM and three read transistors (N_0-N_2). The write operation follows standard SRAM behavior, while the read operation consists of precharge and evaluation phases. During precharge, PCH is asserted, N_0 is turned off, and RBL is precharged to VDD.

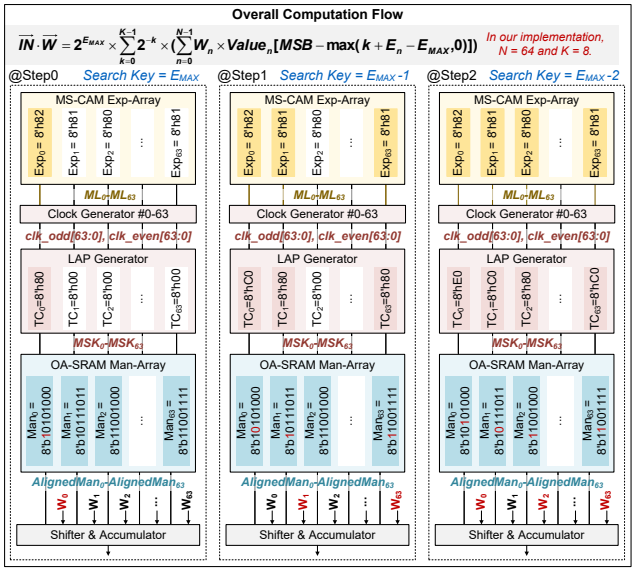
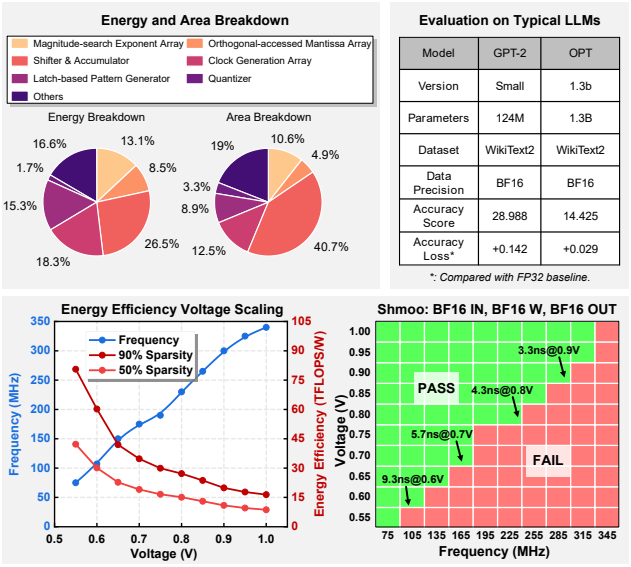


Fig. 8. Overall operation flow of the proposed FP-CIM architecture.

During evaluation, MSK controls cell selection, and RBL is discharged only when $\mathrm{MSK} = \mathrm{Q} = 1$. Fig. 7 (right) illustrates array-level readout organization. Mantissa bits are mapped column-wise from MSB to LSB. With RBL orthogonal to WBL, OA-SRAM outputs one mantissa bit per column per cycle. The one-hot $\mathrm{MSK_i}$ directly indexes the target mantissa bit position without explicit addressing or data movement, enabling alignment to be realized inherently within the memory array. The EVAL subsequently latches the inverted RBL via SA for downstream MAC operations.

Fig. 8 illustrates an example of the overall operation flow of the proposed FP-CIM. In each step, a matched $\mathrm{ML_i}$ in the MS-CAM generate valid pulses through the clock generator, which drive the latch chain in the LAP array to shift the output $\mathrm{TC_i}$ by one bit to the right. For example, in step 2, the first matched $\mathrm{ML_1}$ sets $\mathrm{TC_1} = 8'\mathrm{h80}$, while the second matched $\mathrm{ML_1}$

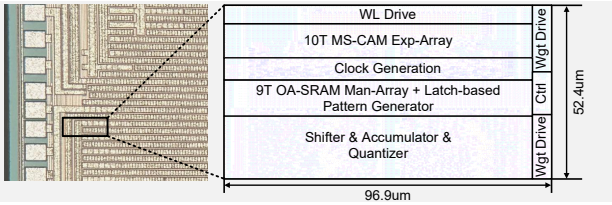


Evaluation on Typical LLMs

Model	GPT-2	OPT
Version	Small	1.3b
Parameters	124M	1.3B
Dataset	WikiText2	WikiText2
Data Precision	BF16	BF16
Accuracy Score	28.988	14.425
Accuracy Loss*	+0.142	+0.029

*: Compared with FP32 baseline.

Fig. 9. Performance evaluation, including breakdown, accuracy results, EF scaling, and shmoo plot.



CHIP SUMMARY			
Technology	12nm FinFET		
Array area	5077.6um²		
Macro size	64 × 16 = 1Kb		
Bitcell Structure*1	10T MS-CAM	9T OA-SRAM	14T LAP Cell
	0.578 um²	0.353 um²	0.637 um²
Supply voltage (V)	0.55 - 1.0		
Access time (ns)	2.9-13.3		
Weight/Input precision (bit)	BF16		
Output precision (bit)	BF16		
Number of Input Channels	64		
Throughput (GFLOPS)	1.2-5.4		
Average energy efficiency*2 (TFLOPS/W)	8.7-42.2		
Peak energy efficiency*3 (TFLOPS/W)	16.4-80.6		
Area efficiency*4 (TFLOPS/mm²)	0.24-1.1		
Model	GPT-2 Small	OPT-1.3B	
Perplexity Loss*5	+0.14 @Wikitext-2	+0.03 @Wikitext-2	

Performance includes all floating-point operations while excluding offline weight alignment. The weight mantissas are pre-aligned to 8 bits.
1. After shrink.
2. Measured at 50% input sparsity, highest average energy efficiency point @VDD=0.55V, 75MHz.
3. Measured at 90% input sparsity, highest peak energy efficiency point @VDD=0.55V, 75MHz.
4. Measured across VDD=0.55-1.0V, with the highest area efficiency achieved @VDD=1.0V, 340MHz.
5. The corresponding software baselines for GPT-2 and OPT-1.3B are 28.846 and 14.396, respectively.

Fig. 10. Chip photograph and summary table.

updates TC_1 from $8'h80$ to $8'hC0$. The TC_i is then converted into a one-hot MSK_i, which guides the OA-SRAM mantissa array to read bits from different rows of each mantissa. In step 2, for instance, RBL_1 reads the first bit, and RBL_0 reads the second bit, thereby achieving implicit mantissa alignment. In this manner, implicit in-memory search-and-index replaces explicit compare-and-shift procedure, enabling efficient FP alignment. Finally, the aligned mantissas (AlignedMan) are multiplied with the input weights, and the results are accumulated in a shift-and-add accumulator to produce 64-channel MAC outputs.

III. Measurement Results

The prototype chip was fabricated in a 12-nm FinFET process. Fig. 9 presents the measured performance and evaluation

	ESSCIRC'22[1]	ISSCC'24[2]	ISSCC'25[3]	ISSCC'25[4]	JSSC'24[5]	This Work
CIM Type	SRAM/CAM-based Digital CIM	ReRAM-based Analog CIM	SRAM-based Digital CIM	SRAM-based Digital CIM	SRAM-based Digital CIM	MS-CAM/OA-SRAM-based Digital CIM
In-memory Comparison	✔	✗	✗	✗	✗	✔
Implicit Alignment	✔	✗	✗	✗	✗	✔
Data Precision	FP8/FP16/FP32	BF16/FP16	INT8/BF16	INT4/INT8/ FP8/BF16	FP16	BF16
Technology	28nm	22nm	28nm	28nm	28nm	12nm
Supply Voltage(V)	0.5-0.9	0.7-0.8	0.55-0.9	0.55-0.9	0.5-1.3	0.55-1.0
Access Time(ns)	2.5-18.9	4.8-6.0	5.4-16.2	2.5-6.5	3.3-80.0	2.9-13.3
Macro Area(mm²)	0.0439	-	0.1360	0.0427	0.2012	0.0051
Energy Efficiency (TFLOPS/W)	1.644-0.099 @(FP8-FP32)	65.5[1] @(BF16)	17.83-62.84[2] @(BF16)	17.2-48 @(BF16)	53.3[3]	16.4-80.6[4]
Area Efficiency (TFLOPS/mm²)	0.06-0.005 @(FP8-FP32)	0.104 @(BF16)	0.14[5] @(BF16)	1.1[5] @(BF16)	0.113	0.24-1.1[6]
Application	CNN @CIFAR-100	CNN @ImageNet	CNN @(ImageNet, COCO)	CNN @CIFAR100	MAC @KITTI	LLM @(Wikitext)

1. Measured @VDD=0.7V and 90% input sparsity. 2. Measured with real case under 0.55V. Real case: 80% input sparsity, 10% input toggle rate, real weights @ResNet50 & ViT. 3. Measured at 50% input sparsity with 50% product skipping, highest peak energy efficiency point @VDD=0.50V, 12.5MHz. 4. Measured at 90% input sparsity, highest energy efficiency point @VDD=0.55V, 75MHz. 5. All results are inferred from the macro area, number of channels, and operating frequency reported in the respective works, and are measured at 0.9 V. 6. Peak performance measured @1.0V, 340MHz.

Fig. 11. Comparison with state-of-the-art FP-CIM macros.

results. Benefiting from lossless BF16 MAC operations, the proposed design achieves perplexities of 28.988 and 14.425 for GPT-2 and OPT-1.3B inference, respectively, with only 0.5% and 0.2% degradation compared to FP32 baselines. The measured shmoo plot shows that the chip operates across a voltage range from 0.55 V to 1.0 V, corresponding to a frequency range of 75-340 MHz. Fig. 10 shows the chip micrograph and summarizes the implementation results. Fig. 11 compares the proposed design with state-of-the-art CIM architectures. Operating at 0.55 V and 75 MHz, the proposed FP-CIM achieves a peak EF of 80.6 TFLOPS/W with BF16 precision, which is 1.2-1.7× higher than prior FP-CIM designs [2–5].

References

[1] S. Jeong et al., "A 28nm 1.644TFLOPS/W floating-point computation SRAM macro with variable precision for deep neural network inference and training," in Proc. IEEE Eur. Solid-State Circuits Conf. (ESSCIRC), 2022, pp. 145–148.

[2] T.-H. Wen et al., "A 22nm 16Mb floating-point ReRAM compute-in-memory macro with 31.2TFLOPS/W for AI edge devices," in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), vol. 67, 2024, pp. 580–582.

[3] X. Wang et al., "14.3 a 28nm 17.83-to-62.84TFLOPS/W broadcast-alignment floating-point CIM macro with non-two's-complement MAC for CNNs and transformers," in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), vol. 68, 2025, pp. 254–256.

[4] 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," in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), vol. 68, 2025, pp. 258–260.

[5] M. Li et al., "SLAM-CIM: A visual SLAM backend processor with dynamic-range-driven-skipping linear-solving FP-CIM macros," IEEE J. Solid-State Circuits, vol. 59, no. 11, pp. 3853–3865, Nov. 2024.

[6] S. Yan et al., "A 28-nm floating-point computing-in-memory processor using intensive-CIM sparse-digital architecture," IEEE J. Solid-State Circuits, vol. 59, no. 8, pp. 2630–2643, Aug. 2024.

[7] 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," in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), 2023, pp. 126–128.

[8] S. Hong et al., "DFX: A low-latency multi-FPGA appliance for accelerating transformer-based text generation," in Proc. IEEE/ACM Int. Symp. Microarchitecture (MICRO), 2022, pp. 616–630.