A 56.2TOPS/W Hybrid-Inner-Outer-Product SRAM Compute-in-Memory Macro for GEMM with Scalable Precision-Adaptive Systolic Dataflow

Wenjie Ren[1], Meng Wu[1], Xiangjun Ye[1], Ruohang Xu[1], Peiyu Chen[2], Ying Liu[1], Fengyun Yan[3],
Jiayoon Ru[1], Yufei Ma[1], Tianyu Jia[1,*] and Le Ye[1,*]
[1]Peking University, Beijing, China; [2]Advanced Institute of Informative Technology of Peking University,
Hangzhou, China; [3]Nano Core Chip Electronic Technology, Hangzhou, China
[*]Corresponding author email: {tianyuj, yele}@pku.edu.cn

***Abstract*—This paper presents an efficient SRAM compute-in-memory (CIM) macro design for general matrix multiplication (GEMM). A hybrid inner-outer product architecture is designed to enable long-time weight reusing and short-time input reusing, reducing 1.57-2.69× overall SRAM access. A data-precision-adaptive dataflow is developed to scale the CIM macros with balanced bandwidth across varying data precisions. Our CIM-friendly dataflow obtains 19-38.5% reduction in operation cycles with the CIM capability of concurrent read/write and computing. Fabricated in 22nm technology, the CIM macro achieves energy efficiency (EE) of 56.2 TOPS/W for GEMM and 71.65 TOPS/W for multiply–accumulate operation (MAC) in 8-bit inputs and weights. The macro and system area efficiency (AE) is 3.18 TOPS/mm^2 and 1.89 TOPS/mm^2, which is 1.42× and 1.12× compared with prior SOTA designs.**

***Index Terms*—Compute-in-Memory, GEMM, scalable dataflow**

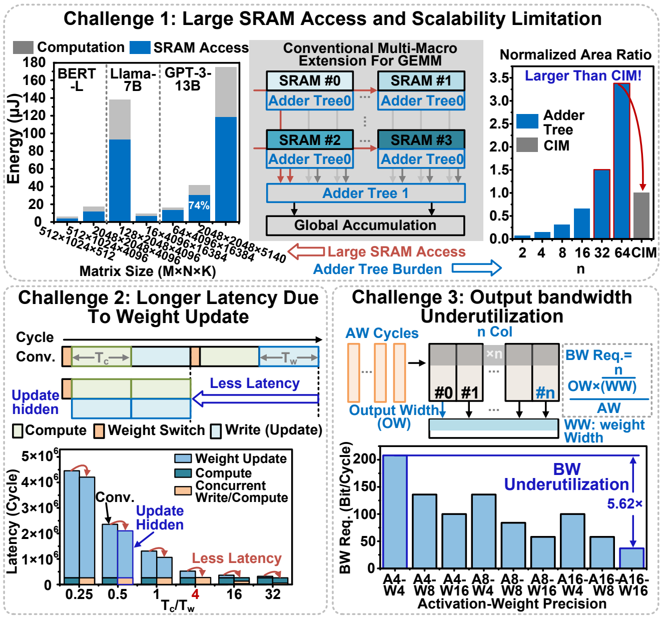


Fig. 1. The design challenges of CIM-based GEMM macro. The latency is estimated for matrix multiplication of 512×512×512. The BW requirement is estimated with a 32×256 DCIM macro.

I. Introduction

Compute-in-memory (CIM) has been widely explored for efficient matrix-vector multiplications (MVM) [1], [2]. However, the general matrix multiplication (GEMM) operation can not be well supported by CIM and is crucial in neural network and consumes significant inference latency/power in emerging natural language models, e.g. >74% in GTP2 [3], [4]. In fact, utilizing CIM for GEMM is non-trivial and presents several new challenges, as shown in Fig. 1. First, the conventional inner-inner product architecture results in substantial external memory access (EMA) for large input matrix (M×K), which can account for up to 74% of total energy consumption. Additionally, the architecture exhibits limited scalability due to multi-level large fan-in adder trees and extensive broadcast paths. As the number of CIM macros scales up, the area overhead of adder tree exceeds the CIM macro area by 3.4×. Second, CIMs suffer from long latency due to the interleaving operations of memory access and computation. Weight update operations of conventional CIM can consume up to 47% of the total operational time, representing a significant performance bottleneck. Third, the output bandwidth (BW) for CIM often remains underutilization across different data precisions. Our analysis based on CIM macros configured with 32 input channels and 256 rows reveals a 5.62× disparity in output bandwidth requirements between A4W4 (4-bit input and 4-bit weight) and A16W16 precision format.

To address the above challenges, we present a digital CIM (DCIM) macro design for GEMM operation (detailed in Fig. 2) with following key features 1) hybrid inner-outer product (HIOP) cores featuring reconfigurable systolic data paths to enhance input and weight reusing among CIM macros, reducing 1.57-2.69× input/output SRAM access. Logic-reuse accumulator is adopted to process CIM output locally to alleviate multi-level adder tree overhead; 2) a Local Cell based parallel DCIM (LC-DCIM) macro that supports concurrent write/read and computation to minimize latency and improve throughput, saving 19%-39.5% operation cycles; 3) precision-adaptive weight switching (PAS) dataflow integrated with multi-cycle quantizer to balance and reduce output BW requirements. Our 22nm CIM macro achieves efficiency 56.2 TOPS/W for GEMM and 71.65 TOPS/W for 8-bit MAC.

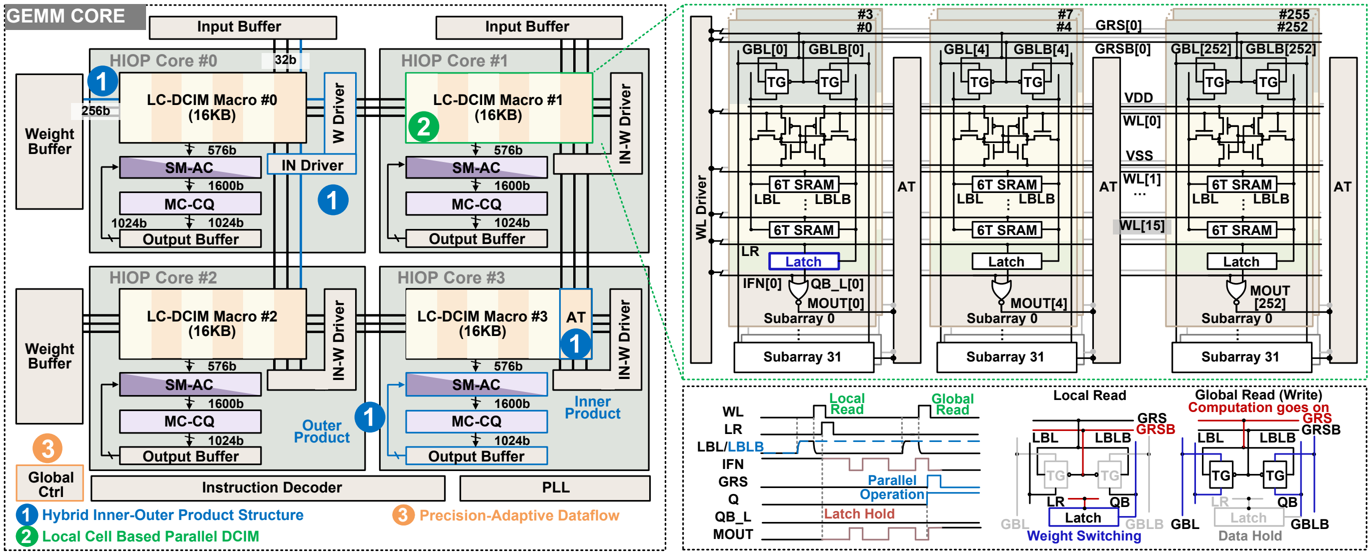


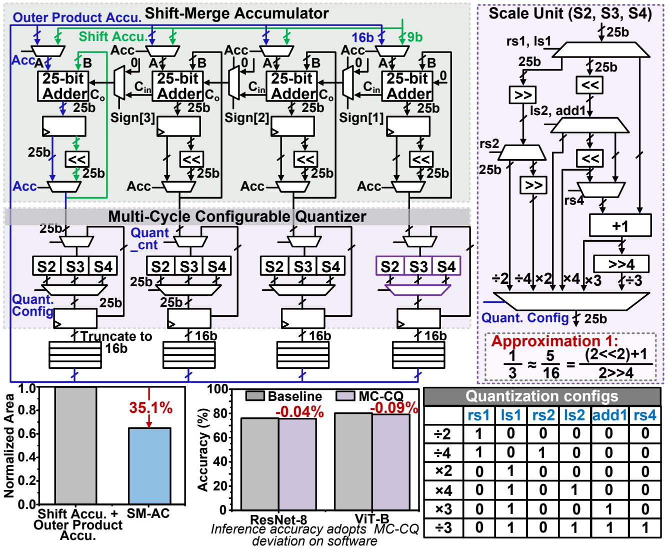
Fig. 2. Overall architecture of our hybrid inner-outer product CIM macro for GEMM operations.

II. Hybrid Inner-Outer Product Structure

Fig. 2 provides an overview of our scalable GEMM CIM design with four HIOP cores. Each HIOP core consists of a 512×256 LC-DCIM macro to support weight update, a shift-merge accumulator (SM-AC), a multi-cycle configurable quantizer (MC-CQ) and a systolic data driver of inputs and weights. The weights can be either pre-loaded or updated into LC-DCIM, while inputs are fetched in bit-serial manner. The weight and input transmit in a systolic or broadcast manner according to mode selection.

Fig. 2 right shows the details of LC-DCIM, which comprises 32×256 CIM cells. Each cell includes a subarray (16 6T SRAMs), a local latch, and a NOR gate. A local latch and two sets of bit lines are adapted to prevent the occupation of SRAM access while computing. When local read (LR) is high, only LBL/LBLB is activated for local reading with transmission gate (TG) closed. The SRAM read-outs are stored in latches for subsequent computation. During global reading, LBL and GBL are both activated and connected by TG. Then, memory port D/Q can write/read SRAM values from GBL/GBLB. The latch retains the historical value for computation when LR is low, enabling concurrent computation and write/read (Fig. 2 waveform). The parallel structure, combined with our dataflow, reduces system latency by 50% as Tc/Tw is 4. LC-CIM generates partial sum (PSUM) in an inner product manner with an adder tree. The PSUMs are then shift accumulated in SM-AC and accumulated the final sum with data from output buffer, implementing the outer product in GEMM. The output is quantized to 16b in MC-CQ within 4 cycles.

Fig. 3 illustrates the details of SA-AC and and MC-CQ in HIOP core. The data path for shift accumulation in inner product is indicated in green while the outer product path is indicated in blue in SM-AC. Four adders are connected in series to enable higher precision addition and are also reused for shift or outer product accumulations, reducing the area by 35.1%. During the quantization cycles, MC-CQ



Quantization configs

	rs1	ls1	rs2	ls2	add1	rs4
÷2	1	0	0	0	0	0
÷4	1	0	1	0	0	0
×2	0	1	0	0	0	0
×4	0	1	0	1	0	0
×3	0	1	0	0	1	0
÷3	0	1	0	1	1	1

Fig. 3. The design of the outer product module, including shift-merge accumulator, multi-cycle configurable quantizer and buffers.

scales the PSUMs by a factor of 2, 3, or 4 in each cycle, depending on the configurations in Fig. 3 bottom. The division operation is implemented as Approximation 1 (Fig. 3 right). With combinations of specific scaling factors, the rescaling of PSUM outputs can be flexible. The accuracy loss is only 0.04% and 0.09% with ResNet-18 on CIFAR-100 and ViT-B on ImageNet-1k using this quantizer for INT8.

III. Precision-Adaptive Systolic Dataflow

Fig. 4 illustrates the reconfigurable systolic data path and the corresponding GEMM mapping strategy across multiple cores. As shown in Fig. 4 left, DCIM macro can internally perform inner products with an adder tree, whereas interactions among HIOP cores compute outer products. Each core independently processes its own PSUMs (e.g., CIM #0 for C_{00}) in a distributed manner resembling output-stationary. This organization alleviates the design overhead of large fan-in

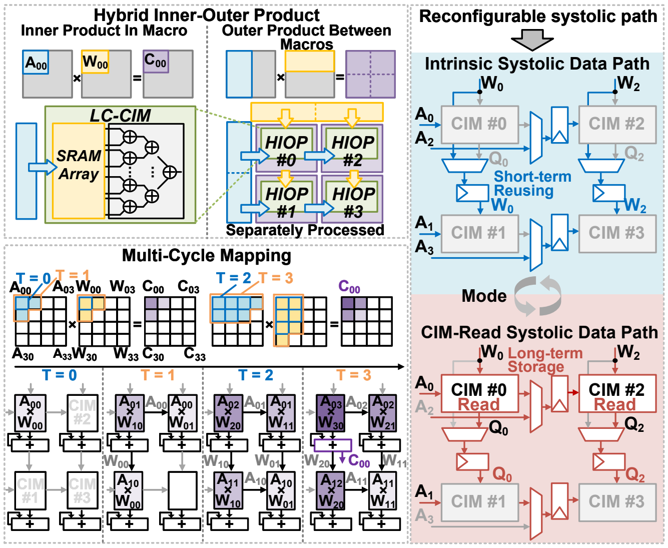


Fig. 4. The structure of reconfigurable systolic path and mapping strategies of hybrid inner-outer product structure for GEMM.

global adder trees, achieving 25.4% area reduction. Fig. 4 right shows the two reconfigurable path for weight exchange between adjacent macros. For intrinsic systolic path, the input and weight are fetched from buffers. Flip-flops are inserted between macros to break long broadcast path in a systolic manner, enhancing the scalability of the multi-CIM design for GEMM computations. For CIM-read data path, the weight data of CIM #1, 3 is fetched from the memory pages of CIM (read-out ports Q) of CIM #0, 2. The CIM subarrays enable long-term weight reusing across adjacent CIM macros while intrinsic systolic path only reuses weight within one period.

Fig. 5 shows our precision-adaptive systolic (PAS) dataflow. This dataflow exploits LC-CIM's concurrent compute and read/write capabilities, hiding the weight sharing and updating under computation. In single HIOP core, the precision-adaptive weight switching is applied to reduce the output bandwidth to output buffer. For an activation with 4b/8b precision (AINT4/8), weight row switches every 4 (8) cycles and 4 (2) shifted PSUMs are accumulated before each storing, reducing BW requirement (BW Req.) by 4.74× and 2.13× (Fig. 5 bottom), respectively. As for AINT16, the shifted PSUMs are directly stored in buffer every 16 cycles. Systolic dataflow is utilized among 4 HIOP cores and two modes are designed to adapt various input channel demands in models. In weight exchange mode, GEMM of large input channels are divided into blocks (Fig. 4) and the weight pages of CIM are shared between adjacent cores by interleaving read and write. The weight exchange can be disabled in reusing mode when the input channel of model is smaller than that of GEMM engine. A critical factor of this dataflow is T_c/T_w, which affects the trade-off between buffer size and system latency (Fig. 5 bottom). T_c and T_w represent the interval between weight updating and the cycle consumed for weight updating. The highest latency reduction of 50% is achieved with relatively small buffer overhead at a balanced T_c/T_w of 4.

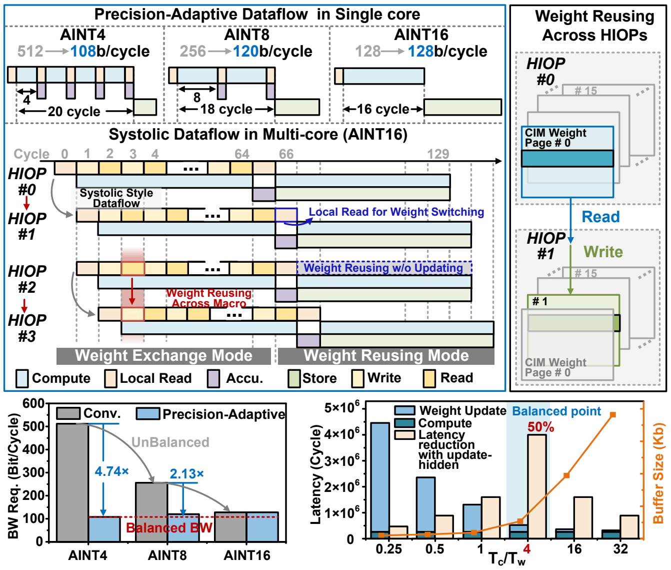


Fig. 5. Precision-adaptive dataflow. The latency is estimated for matrix multiplication of 512×512×512.

IV. Measurement Results

Fig. 6 shows the die photo and the improvement breakdown. Our chip was fabricated in 22nm CMOS technology, and the area is 0.941×0.507 mm^2. The GEMM DCIM macro operates at 0.55-1V with peak frequency of 99.58-880MHz, as shown in Fig. 7. The normalized latency and SRAM acess is evaluated on diverse ML workload dominated by different size of matrix multiplications. The HIOP structure enables the long-time weight reusing and short-time input reusing along reconfigurable systolic path, reducing 1.57-2.69× overall SRAM access. PAS dataflow parallels the weight sharing and updating operation with computation. The HIOP structure and PAS dataflow obtain 19-38.5% reduction in operation cycles compared to the baseline, with negligible accuracy loss.

Fig. 7 (a-b) shows the trend of EE as VDD increased and also details the EE under various weight sparsity and input toggle rates. A peak EE of 56.2 TOPS/W is achieved at 0.55V and 99.58MHz. The AE of GEMM macro reaches 1.89 TOP/mm^2 at 1.00V and 880MHz. Based on practical considerations, Fig. 7 (d) presents effective EE for different matrix sizes. The evaluated matrix size is M×K for matrix A and K×N for matrix W. As M (N=M) varies with K fixed at 64, the effective EE increases as matrix size increases, which attributes to longer weight reusing. As K increases with M(N) fixed at 64, the effective EE is nearly unchanged as weight updating is frequently required and the reusing rates are relatively constant. Our GEMM macro demonstrates an increasing effective EE as larger matrices are processed, adapting to the demands of larger models (e.g., LLMs).

Fig. 8 compare our chip with prior SOTA CIMs [5]–[8]. Our work adopts a hybrid inner-outer product structure to reduce the power-consuming SRAM access. The GEMM core achieves the best energy efficiency of 71.65 TOPS/W and 56.2 TOPS/W for MAC operation at CIM macro level and GEMM operation at system level. The AE of 3.18 TOPS/mm^2 of CIM

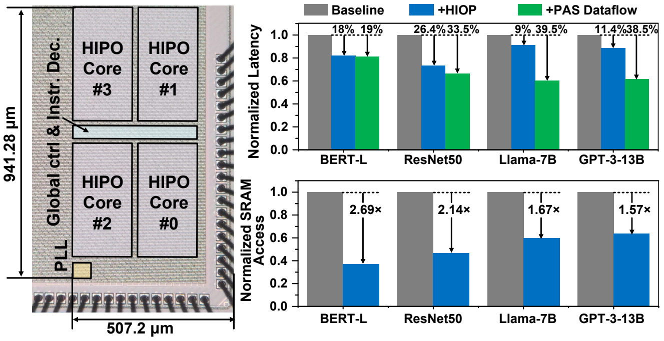


Fig. 6. Chip micrograph and improvement breakdown.

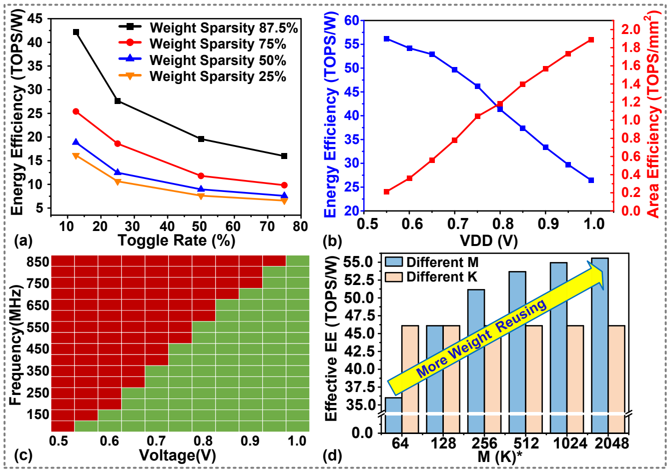
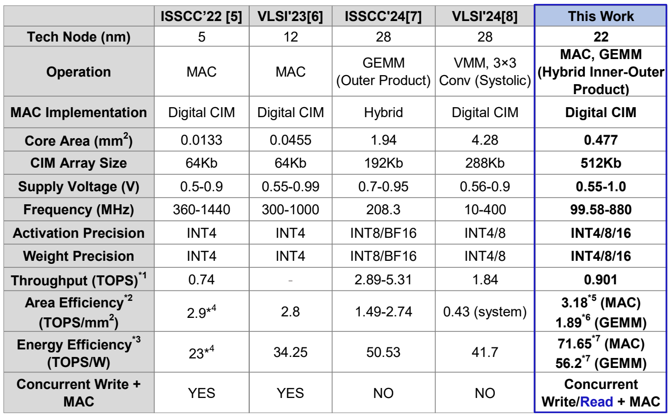


Fig. 7. (a) The measurement results of energy efficiency at different weight sparsity and input toggle rates, (b) the energy efficiency and area efficiency at different supply voltage, (c) shmoo plot, (d) the effective energy efficiency for different matrix sizes.

macro is the best among SOTA CIM macros [5], [6], [8], with [5] is normalized from 5nm to 22nm based on the scale factor in [9]. Our DCIM further supports concurrent write/read and computation to hide weight updating latency in PAS dataflow, enabling 19-39.5% reduction in overall latency.

V. Conclusion

In this work, we introduce a scalable SRAM compute-in-memory (CIM) macro for general matrix multiplication (GEMM). Our design adopts a hybrid inner-outer product architecture to enhance input and weight reusing and reduce the overhead of large-scale adder trees across macros. A local latch and two sets of bit lines are adapted in LC-CIM to support concurrent read/write and computing. The data-precision-adaptive dataflow balances output bandwidth across varying data precisions and parallels the weight-update. This design reduces the overall SRAM access by 1.57-2.69× and obtains 19-38.5% reduction in operation cycles. Fabricated in 22nm technology, the overall design achieves energy efficiency (EE) of 56.2 TOPS/W for GEMM and 71.65 TOPS/W for MAC in 8-bit inputs and weights. The macro and system area efficiency is 3.18 TOPS/mm^2 and 1.89 TOPS/mm^2.



	ISSCC'22 [5]	VLSI'23[6]	ISSCC'24[7]	VLSI'24[8]	This Work
Tech Node (nm)	5	12	28	28	**22**
Operation	MAC	MAC	GEMM (Outer Product)	VMM, 3×3 Conv (Systolic)	**MAC, GEMM (Hybrid Inner-Outer Product)**
MAC Implementation	Digital CIM	Digital CIM	Hybrid	Digital CIM	**Digital CIM**
Core Area (mm^2)	0.0133	0.0455	1.94	4.28	**0.477**
CIM Array Size	64Kb	64Kb	192Kb	288Kb	**512Kb**
Supply Voltage (V)	0.5-0.9	0.55-0.99	0.7-0.95	0.56-0.9	**0.55-1.0**
Frequency (MHz)	360-1440	300-1000	208.3	10-400	**99.58-880**
Activation Precision	INT4	INT4	INT8/BF16	INT4/8	**INT4/8/16**
Weight Precision	INT4	INT4	INT8/BF16	INT4/8	**INT4/8/16**
Throughput (TOPS)^{*1}	0.74	-	2.89-5.31	1.84	**0.901**
Area Efficiency^{*2} (TOPS/mm^2)	2.9^{*4}	2.8	1.49-2.74	0.43 (system)	**3.18^{*5} (MAC) 1.89^{*6} (GEMM)**
Energy Efficiency^{*3} (TOPS/W)	23^{*4}	34.25	50.53	41.7	**71.65^{*7} (MAC) 56.2^{*7} (GEMM)**
Concurrent Write + MAC	YES	YES	NO	NO	**Concurrent Write/Read + MAC**

*1 One operation (OP) = one multiplication or one addition, normalized to 8bit
*2 Measured at 1.0V, 880MHz. *3 Measured at 0.55V, 99.58MHz.
*4 Normalized to 22nm with the scale factor of 19× and 2.8× as [9]
*5 Evaluating only CIM Macro for INT8 MAC operation. *6: Full chip evaluation;
*7 At 12.5% input toggle rate and 87.5% weight bit = 0 distribution.

Fig. 8. Comparison table.

Acknowledgment

This work is supported by NSFC No. 92164301, 92464301, 62225401, U24A20287.

References

[1] H. Fujiwara *et al.*, "34.4 A 3nm, 32.5TOPS/W, 55.0TOPS/mm^2 and 3.78Mb/mm^2 Fully-Digital Compute-in-Memory Macro Supporting INT12 × INT12 with a Parallel-MAC Architecture and Foundry 6T-SRAM Bit Cell," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, (San Francisco, CA, USA), pp. 572–574, IEEE, Feb. 2024.

[2] 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," in *2023 IEEE International Solid- State Circuits Conference (ISSCC)*, (San Francisco, CA, USA), pp. 128–130, IEEE, Feb. 2023.

[3] S. Kim *et al.*, "20.5 C-Transformer: A 2.6-18.1J/Token Homogeneous DNN-Transformer/Spiking-Transformer Processor with Big-Little Network and Implicit Weight Generation for Large Language Models," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, (San Francisco, CA, USA), pp. 368–370, IEEE, Feb. 2024.

[4] R. Karami *et al.*, "NonGEMM Bench: Understanding the Performance Horizon of the Latest ML Workloads with NonGEMM Workloads," Mar. 2025. arXiv:2404.11788 [cs].

[5] H. Fujiwara *et al.*, "A 5-nm 254-TOPS/W 221-TOPS/mm ^{2} Fully-Digital Computing-in-Memory Macro Supporting Wide-Range Dynamic-Voltage-Frequency Scaling and Simultaneous MAC and Write Operations," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, (San Francisco, CA, USA), pp. 1–3, IEEE, Feb. 2022.

[6] G. Jedhe *et al.*, "A 12nm 137 TOPS/W Digital Compute-In-Memory using Foundry 8T SRAM Bitcell supporting 16 Kernel Weight Sets for AI Edge Applications," in *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, (Kyoto, Japan), pp. 1–2, IEEE, June 2023.

[7] 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," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, (San Francisco, CA, USA), pp. 576–578, IEEE, Feb. 2024.

[8] Z. Dai *et al.*, "A 41.7TOPS/W@INT8 Computing-in-Memory Processor with Zig-Zag Backbone-Systolic CIM and Block/Self-Gating CAM for NN/Recommendation Applications," in *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, (Honolulu, HI, USA), pp. 1–2, IEEE, June 2024.

[9] 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 *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64, pp. 252–254, Feb. 2021. ISSN: 2376-8606.