14.2 A 16nm 216kb, 188.4TOPS/W and 133.5TFLOPS/W Microscaling Multi-Mode Gain-Cell CIM Macro Edge-AI Devices

Win-San Khwa*[1], Ping-Chun Wu*[2], Jian-Wei Su[2,3], Chiao-Yen Cheng[2], Jun-Ming Hsu[2], Yu-Chen Chen[2], Le-Jung Hsieh[2], Jyun-Cheng Bai[2], Yu-Sheng Kao[2], Tsung-Han Lou[2], Ashwin Sanjay Lele[4], Jui-Jen Wu[1], Jen-Chun Tien[2], Chung-Chuan Lo[2], Ren-Shuo Liu[2], Chih-Cheng Hsieh[2], Kea-Tiong Tang[2], Meng-Fan Chang[1,2]

[1]TSMC Corporate Research, Hsinchu, Taiwan
[2]National Tsing Hua University, Hsinchu, Taiwan
[3]Industrial Technology Research Institute, Hsinchu, Taiwan
[4]TSMC Corporate Research, San Jose, CA

*Equally Credited Authors (ECAs)

The evolution of AI workloads demands ever higher resolutions, multiple data-format compatibility, and high-energy efficiency (EEF). Integer (INT) CIM macros [1-13,24] enable high-EEF MAC operations at the cost of limited accuracy. Floating-point (FP) CIMs [14-21] provide higher accuracy but at the cost of reduced EEF due to a higher bit-width and complex pre- and post-operation normalization. Emerging microscaling (MX) data formats [22-23] improve the tradeoff between accuracy and EEF by employing a shared-scale (SS) factor across *k*-elements within a block; thereby, allowing precise control over the dynamic range and resolution with a low bit-width for high EEF. Application-level metrics indicate that the optimal data format is workload-dependent, necessitating novel CIM macros that integrate MX hardware with FP-INT processing. As shown in Fig. 14.2.1, CIMs face challenges in achieving high-EEF multi-mode (MX-INT-FP) MAC operations; due to (1) high-energy consumption and system area overhead from system-to-CIM data transfers, including FP-to-MX conversion (FP2MX) and SS processing outside of the CIM; (2) significant area and energy consumption by the adder tree (ADT); and (3) inefficient data transfers along a single write path, i.e. to CIM memory (MEM) or ALU buffer (BUF), for various accumulation sizes (N_{CH}) across multiple layers of neural-network (NN) models. This paper presents an MX-INT-FP multi-mode CIM (M2-CIM) featuring: (1) a multi-mode input (IN) processing unit (M2-IPU) with an SS-variance-aware MAC flow (SS-VAF), which enables both FP2MX and SS processing within the CIM to eliminate system-to-CIM data transfers and reduce compute energy; (2) a pattern-aware hybrid adder tree (PAH-ADT) to reduce energy consumption for high-probability IN-patterns using a smaller area with higher EEF and area efficiency (AEF); and (3) an accumulation-aware data flow (A2-DF), supporting a dynamically reconfigurable data write path and a compute flow based on the workload N_{CH} to enhance hardware flexibility and reduce data-transfer energy consumption. A 16nm 216kb MX-INT-FP multi-mode gain-cell (GC) CIM is fabricated using compact-area 3.x transistors (3.xT) GCs. This design supports MX-MACs with 128 accumulations (N_{ACCU}) for MXFP8/MXINT8 INs and weights (W), which generate FP32 outputs (OUT). Also supported are FP-MAC operations with N_{ACCU}=128/64/64 for FP8/FP16/BF16 IN and W and FP32-OUT, as well as INT-MACs with N_{ACCU}=128 for 4/8b IN and W, generating 15/23b-OUT. This M2-CIM achieves an MX-MAC EEF of 133.5TFLOPS/W in MXINT8 mode and an FP-MAC EEF of 91.9TFLOPS/W in BF16 mode.

Figure 14.2.2 illustrates the proposed M2-IPU and SS-VAF. Existing INT-CIM and FP-CIM macros lack support for SS processing and FP2MX for MX-MAC operations, resulting in large system-to-CIM data transfer due to MX-related computations outside of the CIM. M2-IPU incorporates MX-related functions within the CIM by reusing the FP exponent (EXP) processing circuit as both an SS pre-processing circuit and an FP2MX converter; thereby, reducing data transfers and enhancing EEF and AEF. Our M2-CIM consists of 24 banks, each with an OUT channel. Each bank includes an M2-IPU, a multi-mode computing array (M2-CA), a PAH-ADT, a post-processing block (PO-PRO), and an OUT-processing block (OUT-PRO). The M2-IPU hardware comprises of a maximum finder (MAX-FD), a product (PD) SS variance detector (PD-VD), a hybrid-align-number generator (HAL-GEN), SS controller (CTRL) and IN-mantissa (IN_M) processing unit (INM-PU). Most layers in NN models exhibit a high variance (HV) in channel-wise PD distributions; with a maximum PD value that is significantly larger than others. When channel-wise PDs are accumulated into a MAC value (MACV), the LSBs of the smaller PDs may be truncated during conversion of the MACV OUT to FP32 format, due to the HV distribution. The proposed SS-VAF leverages variance information generated during the SS pre-processing phase to enhance EEF, without accuracy loss, by increasing IN_M sparsity when processing MX-MAC. The SS-VAF operation is implemented in four phases. In phase 0 (Ph0), the M2-IPU pre-processes full-channel SS computations and identifies the variance in channel-wise PDs, based on the maximum-PD-SS ($PD_{SS\text{-}MAX}$) and the SS threshold ($PD_{SS\text{-}TH}$). The $PD_{SS\text{-}TH}$ value is a pre-trained parameter used to prevent accuracy loss in the target application. If $PD_{SS\text{-}MAX} > PD_{SS\text{-}TH}$, then the M2-CIM operates in HV mode for the subsequent full-channel MAC operation; otherwise, it operates in low-variance (LV) mode. In phase 1 (Ph1), the M2-CIM performs INT- and FP-MAC operations based on the mode identified in Ph0 and outputs an FP32 MACV. In LV mode, the M2-IPU aligns IN_M based on the difference in PD-EXP (ΔPD_E) and the PO-PRO aligns block-wise partial MACVs (pMACV) based on their difference in PD_{SS} (ΔPD_{SS}) without loss of accuracy. In HV mode, the M2-IPU aligns IN_M based on both ΔPD_E and ΔPD_{SS}, with extra shifting in IN_M from ΔPD_{SS}, which increases IN_M sparsity, further enhancing EEF. In phase 2 (Ph2), the OUT-PRO processes the activation function of the M2-CIM outputs and generates INs for the next layer. In phase 3 (Ph3), the M2-IPU converts FP INs of the subsequent layer to MX format. Note that macro's AEF is enhanced by reusing the M2-IPU as the SS pre-processing circuit in Ph0, as the EXP processing circuit in Ph1, and as the FP2MX converter in Ph3.

Figure 14.2.3 illustrates the structure and operation of the PAH-ADT. A conventional 28T ADT [27] provides a standard OUT at the cost of a large area and power overhead. Whereas, a 14T ADT [26] has a smaller area but a longer compute latency due to its serially-connected 14T full adders (FAs) with a gate-diffusion IN and an insufficient signal-driving capability. A hybrid ADT [24], which interleaves 14 and 28T FAs, improves the tradeoff between area and latency by mitigating signal-drive issues. The proposed PAH-ADT further improves EEF and AEF by integrating 11, 14, 15 and 28T FAs. This software-hardware co-design reduces energy consumption and the area of the ADT for high-probability IN patterns. In some NN models, high place-value (PV) ADT FAs frequently encounter IN patterns A/B/Ci, such as 0/0/0 and 0/1/0. Conventional 14 and 28T FAs include redundant transistors (MB, MC, and MD) to ensure a standard OUT for less critical IN patterns. The proposed pattern-aware 11 and 15T FAs remove these redundant transistors to reduce the FA area and optimize its energy consumption for high-probability patterns. Ultra-low V_t devices minimize compute latency, even when operating at low V_{DD}. The PAH-ADT improves macro EEF and AEF by replacing the original 14 and 28T FAs with pattern-aware 11 and 15T FAs in the high-PV ADT piece.

Figure 14.2.4 illustrates the proposed A2-DF and 3.xT GC. The layers in NNs often differ in N_{CH}, complicating data transfers using a single write path to the memory or an ALU buffer. When the required N_{CH} exceeds the N_{ACCU} that the CIM can process in a single pass (N_{CIM}), the computation must be divided into multiple slices; thereby, necessitating repeated data fetches from the system W buffer. Conventional CIM macros [2] with only an ALU buffer write path requires repeated system-to-CIM data transfers and additional area for intermediate data storage. When a CIM can handle the N_{CH} in one pass ($N_{CIM} \geq N_{CH}$), the hardware constrained to a memory write path [15] suffers low EEF, due to the redundant movement of internal CIM data. A2-DF features a reconfigurable dual-path data-write mechanism that dynamically adjusts the write path based on the N_{CH} to enhance hardware flexibility and reduce energy consumption. Conventional designs rely on single-path-write IO circuits with a high area overhead; however, A2-DF integrates the write-IO function into the IN driver: reducing area while supporting both stationary (ALU buffer) and non-stationary (memory) write paths. In scenarios where $N_{CIM} < N_{CH}$, A2-DF employs a non-stationary write path to the memory array, minimizing repetitive data transfers from buffer to CIM, and reduces intermediate storage, compared to an implementation with only an ALU-buffer write path. In scenarios where $N_{CIM} \geq N_{CH}$, A2-DF employs a stationary write path direct to the ALU buffer, bypassing the memory array to reduce energy consumption, compared to an implementation with only a memory write path. The proposed 3.xT GC improves data-retention time 3× by incorporating an N-PODE three-terminal device, instead of a GC without an N-PODE device. This device increases the parasitic capacitance of the GC storage node without additional area overhead, thereby enhancing macro performance.

Figure 14.2.5 summarizes the performance of the various schemes. M2-CIM reduces data transfers by 72-85%, compared to prior CIMs without M2-IPU and A2-CF, by enabling MX-MAC operations within the CIM and incorporating a reconfigurable dual-path write. PAH-ADT improves ADT's FoM1 by 1.43-2.79×, compared to prior designs [24-25] when applying MobileViT to ImageNet, due to its optimized area and computational power for high-probability FA-IN patterns. The EEF-area-accuracy FoM2 of M2-CIM is 1.4-2.7× higher than prior CIMs [14-16] when applying ResNet18 to ImageNet, due to its low energy consumption and macro area enabled by the MX-MAC within the CIM and the proposed enhancements.

Figure 14.2.6 presents measured results from a 16nm 216kb GC-CIM for MX-MAC (MXINT8-IN and MXINT8-W with N_{ACCU} = 128 and FP32-OUT) and FP-MAC (BF16-IN and BF16-W with N_{ACCU} = 64 and FP32-OUT) operations. Shmoo plots confirm that the M2-CIM achieves a 3.8ns T_{MAC} for MX-MAC and FP-MAC at V_{DD}=0.8V. This M2-CIM achieves an FoM (OUT-ratio × Normalized EEF × Normalized AEF) that is 1.95-5.73× higher than prior CIMs. The system-level inference accuracy in MXINT8 mode is only 0.05% lower than a FP32 software-based model, when applying ResNet-20 to CIFAR-100. In BF16 mode, it is only 0.02% lower, when applying ResNet-18 to ImageNet. Figure 14.2.7 presents the die photo.

Acknowledgement:
The authors thank Philip Wong, Kerem Akarvardar, Bo Zhang, Ping-Sheng Wu, Xiaoyu Sun, Xiaochen Peng, Brian Crafton, and TSMC colleagues for their guidance, and for financial support from NSTC and TSMC-NTHU Major League.

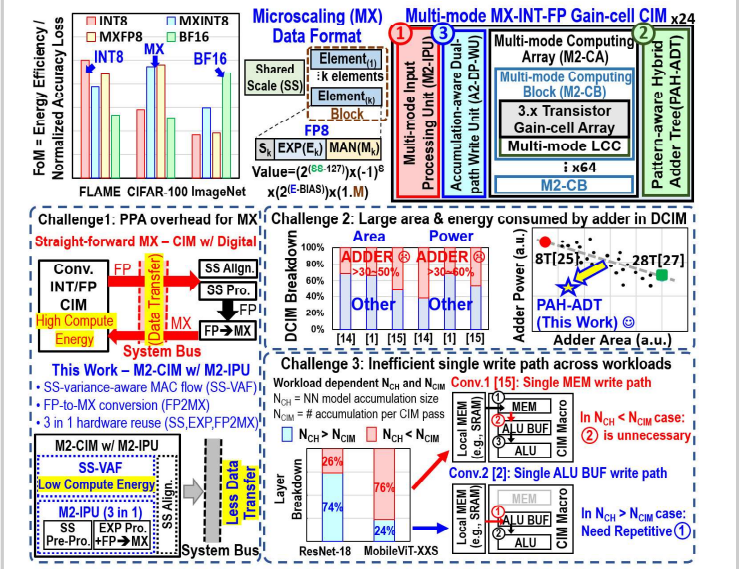


Figure 14.2.1: Design challenges in AI hardware design and the proposed multi-mode MX-INT-FP gain-cell-CIM structure.

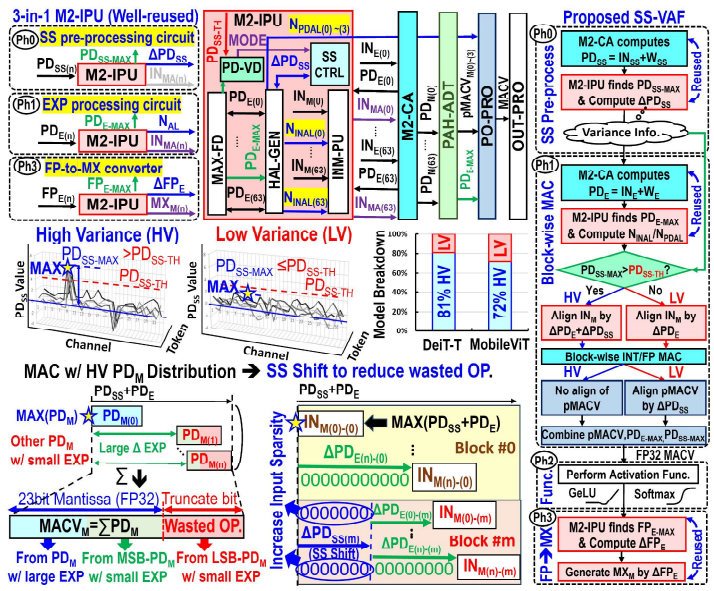


Figure 14.2.2: M2-IPU and SS-variance-aware MAC flow (SS-VAF) operation.

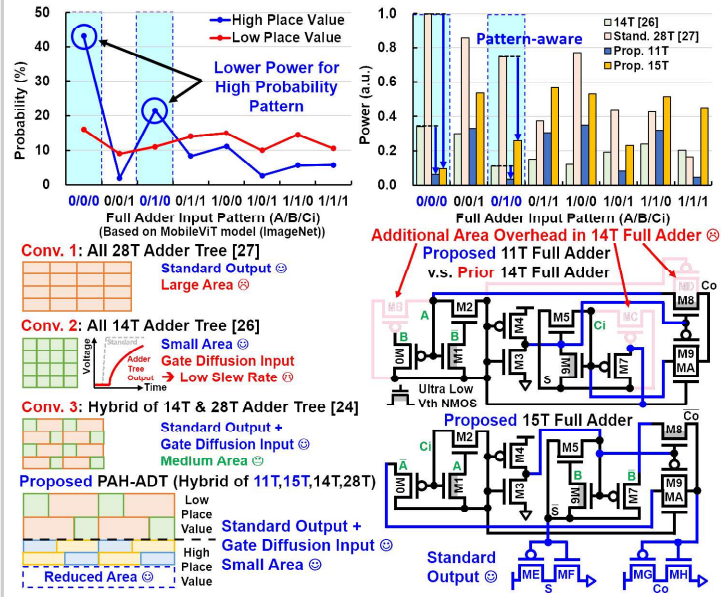


Figure 14.2.3: PAH-ADT structure and operation.

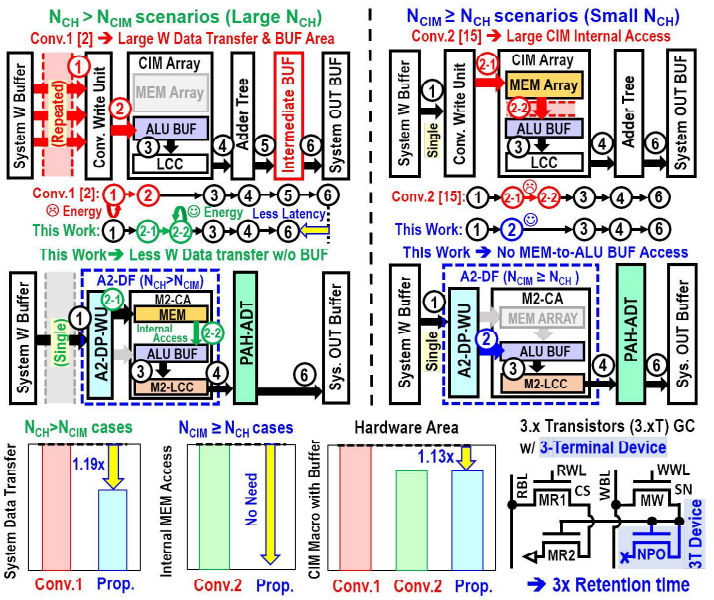


Figure 14.2.4: A2-DF and 3.x-transistor gain-cell operation.

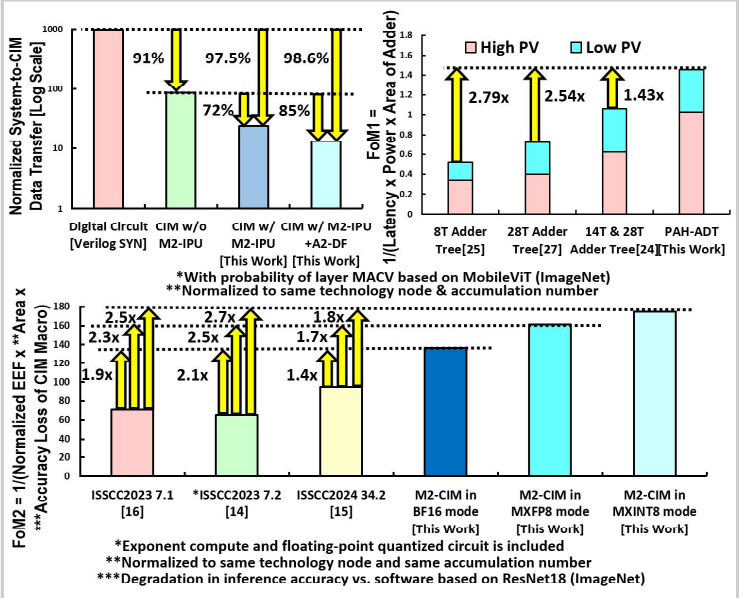


Figure 14.2.5: Simulated performance of the proposed schemes.

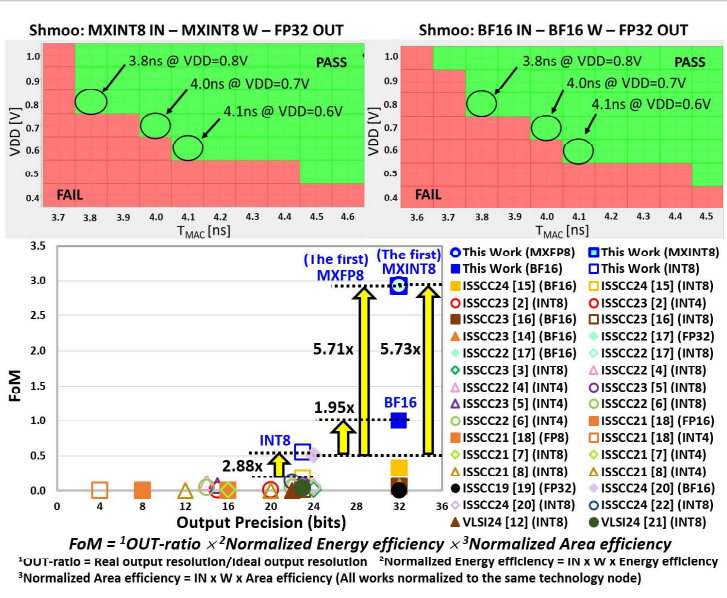
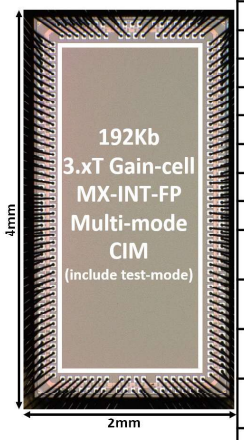


Figure 14.2.6: Measurement results.



CHIP SUMMARY			
Technology	16nm FinFET CMOS		
Macro size	216Kb		
Input precision (bit)	INT8	BF16	MXINT8
Weight precision (bit)	INT8	BF16	MXINT8
Number of IN channels (N_{ACCU})	128	64	128
Number of OUT channels	24	24	24
Output precision (bit)	INT23	FP32	FP32
Supply voltage (V)	0.4-0.8		
Access time (ns) (0.8V)	1.8	3.8	3.8
AVG. Energy efficiency (0.4V) (TOPS/W) (TFLOPS/W)	137.92	63.14	93.11
[1]Peak Energy efficiency (0.4V) (TOPS/W) (TFLOPS/W)	188.41	91.92	133.51
Area efficiency (0.8V) ($TOPS/mm^2$)($TFLOPS/mm^2$)	23.9	5.75	11.4
[2]Inference Accuracy Loss (ResNet-20, CIFAR-100)	*-0.19%	*-0.01%	*-0.05%
[3]Inference Accuracy Loss (ResNet-18, ImageNet)	*-0.55%	*-0.02%	*-0.39%

[1]Performance in 90% input sparsity, 10% input toggle rate (ResNet-20)
[2]Software baseline (FP32) was 70.22% [3]Software baseline (FP32) was 68.31%
*Except for inherent loss from format, no additional loss is introduced by the hardware

Figure 14.2.7: Die photo and performance summary table.

References:
[1] Y. He et al., "A 28nm 2.4Mb/mm^2 6.9 - 16.3TOPS/mm^2 eDRAM-LUT-Based Digital-Computing-in-Memory Macro with In-Memory Encoding and Refreshing," *ISSCC*, pp. 578-580, 2024.
[2] B. Wang et al., "A 28nm Horizontal-Weight-Shift and Vertical-feature-Shift-Based Separate-WL 6T-SRAM Computation-in-Memory Unit-Macro for Edge Depthwise Neural-Networks," *ISSCC*, pp. 134-136, 2023.
[3] H. Mori et al., "A 4nm 6163-TOPS/W/b 4790-TOPS/mm^2/b SRAM Based Digital-Computing-in-Memory Macro Supporting Bit-Width Flexibility and Simultaneous MAC and Weight Update," *ISSCC*, pp. 132-134, 2023.
[4] 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," *ISSCC*, pp. 186-188, 2022.
[5] S.-E. Hsieh et al., "A 70.85-86.27TOPS/W PVT-Insensitive 8b Word-Wise ACIM with Post-Processing Relaxation," *ISSCC*, pp. 136-138. 2023.
[6] P.-C. Wu et al., "A 28nm 1Mb Time-Domain Computing-in-Memory 6T-SRAM Macro with a 6.6ns Latency, 1241GOPS and 37.01TOPS/W for 8b-MAC Operations for Edge-AI Devices," *ISSCC*, pp. 190-192, 2022.
[7] Y.-D. Chih et al., "An 89TOPS/W and 16.3TOPS/mm^2 All-Digital SRAM-Based Full-Precision Compute-In Memory Macro in 22nm for Machine-Learning Edge Applications," *ISSCC*, pp. 252-254, 2021.
[8] J.-W. Su et al., "A 28nm 384kb 6T-SRAM Computation-in-Memory Macro with 8b Precision for AI Edge Chips," *ISSCC*, pp. 250-252. 2022.
[9] B. Yan et al., "A 1.041-Mb/mm^2 27.38-TOPS/W Signed-INT8 Dynamic-Logic-Based ADC-less SRAM Compute-in-Memory Macro in 28nm with Reconfigurable Bitwise Operation for AI and Embedded Applications," *ISSCC*, pp. 188-190, 2022.
[10] P. Chen et al., "A 22nm Delta-Sigma Computing-In-Memory ($\Delta\sum$CIM) SRAM Macro with Near-Zero-Mean Outputs and LSB-First ADCs Achieving 21.38TOPS/W for 8b-MAC Edge AI Processing," *ISSCC*, pp. 140-142, 2023.
[11] H. Fujiwara et al., "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," *ISSCC*, pp. 572-574, 2024.
[12] Y.-J. Lee et al., "34.7A/mm^2 Scalable Distributed All-Digital 6×6 Dot-LDOs Featuring Freely Linkable Current-Sharing Network: A Fine-Grained On-Chip Power Delivery Solution in 28nm CMOS," *ISSCC*, pp. 272-274, 2024.
[13] S Hong et al., "Dyamond: A 1T1C DRAM In-memory Computing Accelerator with Compact MAC-SIMD and Adaptive Column Addition Dataflow," *IEEE VLSI*, pp.1-2, June 2024.
[14] 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," *ISSCC*, pp. 128-129, 2023.
[15] W.-S. Khwa et al., "A 16nm 96Kb Integer/Floating-Point Dual-Mode-Gain-Cell-Computing-in-Memory Macro Achieving 73.3-163.3TOPS/W and 33.2-91.2TFLOPS/W for AI-Edge Devices," *ISSCC*, pp. 568-570, 2024.
[16] P.-C. Wu et al., "A 22nm 832kb Hybrid-Domain Floating-Point SRAM In-MemoryCompute Macro with 16.2-70.2TFLOPS/W for High-Accuracy AI-Edge Devices," *ISSCC*, pp. 126-127, 2023.
[17] F. Tu et al., "A 28nm 29.2TFLOPS/W BF16 and 36.5TOPS/W INT8 Reconfigurable Digital CIM Processor with Unified FP/INT Pipeline and Bitwise In-Memory Booth Multiplication for Cloud Deep Learning Acceleration," *ISSCC*, pp. 254-256, 2022.
[18] A. Agrawal et al., "A 7nm 4-Core AI Chip with 25.6TFLOPS Hybrid FP8 Training, 102.4TOPS INT4 Inference and Workload-Aware Throttling," *ISSCC*, pp. 144-146, 2021.
[19] J. Wang et al., "A Compute SRAM with Bit-Serial Integer/Floating-Point Operations for Programmable In-Memory Vector Acceleration," *ISSCC*, pp. 224-226, 2019.
[20] Y. Yuan et al., "A 28nm 72.12TFLOPS/W Hybrid-Domain Outer-Product Based Floating-Point SRAM Computing-in-Memory Macro with Logarithm Bit-Width Residual ADC," *ISSCC*, pp. 576-578, 2024.
[21] D.-Q. You et al., "A 22nm Nonvolatile AI-Edge Processor with 21.4TFLOPS/W using 47.25Mb Lossless-Compressed-Computing STT-MRAM Near-Memory-Compute Macro," *IEEE VLSI*, pp. 1-2, June 2024.
[22] B. D. Rouhani et al., "Microscaling data formats for deep learning," *arXiv preprint arXiv:2310.10537*, 2023.
[23] B. D. Rouhani et al., "With Shared Microexponents,A Little Shifting Goes a Long Way," *Proceedings of the 50th Annual International Symposium on Computer Architecture,* pp. 1-13, June 2023.
[24] Y.-D. Chih et al., "An 89TOPS/W and 16.3TOPS/mm^2 All-Digital SRAM-Based Full-Precision Compute-In Memory Macro in 22nm for Machine-Learning Edge Applications," *ISSCC*, pp. 252-254, 2021.
[25] Y. Wei et al., "Design of a novel low power 8-transistor 1-bit full adder cell," *Journal of Zhejiang University SCIENCE*C, pp. 604-607, 2011.
[26] O. A. Badry et al., "Low power 1-Bit full adder using Full-Swing gate diffusion input technique," *ITCE*, pp. 205-208, 2018.
[27] V. Gupta et al., "Low-Power Digital Signal Processing Using Approximate Adders," *IEEE TCAD*, vol. 32, no. 1, pp. 124-137, 2013.

 979-8-3315-4101-9/25/$31.00

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

Xing Wang*[1,2], Tianhui Jiao*[1], Yi Yang[1], Shaochen Li[1], Dongqi Li[1], An Guo[1], Yuhui Shi[1], Yuchen Tang[1], Jinwu Chen[1], Zhican Zhang[1], Zhichao Liu[1], Bo Liu[1], Weiwei Shan[1], Xin Wang[3], Hao Cai[1], Wenwu Zhu[3], Jun Yang[1,2], Xin Si[1]

[1]Southeast University, Nanjing, China
[2]National Center of Technology Innovation for EDA, Nanjing, China
[3]Tsinghua University, Beijing, China

*Equally Credited Authors (ECAs)

The rapid advancement of artificial-intelligence (AI) models has increased demand for high-precision and energy-efficient edge-AI chips. Floating-point (FP) support is essential for high-precision neural-network (NN) training and inference; yet FP incurs higher energy and area overhead due to complex FP multiplication and accumulation (MAC) operations. Digital compute-in-memory (DCIM) and floating-point CIM (FP-CIM) [1-10] have emerged as promising techniques to improve energy efficiency with higher accuracy. Previous FP-CIM implementations [1-7] achieved good performance through various alignment schemes and computing processes. However, as illustrated in Figure 14.3.1, the implementation of a digital-domain FP-CIM faces several challenges: (1) the difficulty of balancing FP-computation precision and input reusability, as alignment operations are unfriendly to CIM structure; (2) a large performance loss or area overhead due to peripheral parallel-alignment schemes; and (3) huge digital-MAC dynamic-energy consumption due to low 2's-complement (2C) negative-weight sparsity, coupled with an additional sign-bit computation overhead in digital CIM. This work presents a hierarchical broadcast-alignment non-2's-complement-MAC (B-A-N2CMAC) FP-CIM macro, featuring (1) a broadcast input and embedded lightweight convertor structure to enable BF16/INT8 MAC operations with an improved input reusability; (2) an embedded area-efficient adaptive-alignment scheme with a dual-bit serial MAC; and (3) a format-mixed N2CMAC to reduce dynamic circuit activity and signed computation overhead. A 28nm 64kb B-A-N2CMAC FP-CIM macro is fabricated to support FP-MAC operations using BF16 and INT8 representations. This CIM macro achieved an energy efficiency of 62.84TFLOPS/W for BF16 and 90.15TOPS/W for INT8.

Figure 14.3.2 illustrates the overall structure of the proposed B-A-N2CMAC FP-CIM macro, which includes 16 triple-stacked arrays (TS-As), a hierarchical input buffer, a calibrated accumulator and quantizer (CA&Q), and peripheral circuits. Each TS-A computes one output channel: a TS-A comprises of (1) an embedded serial-input converting unit (ESICU), (2) a 16×256 6T SRAM array, and (3) a format-mixed non-2's-complement MAC unit (N2CMACU). Unlike previous FP-CIM designs [2,3], global floating-point or integer inputs (G_{IN}), from the hierarchical input buffer, can be reused by 16 TS-As to maximize the array-utilization rate. Each G_{IN} consists of an 8b parallel-exponent input (E_{IN}) and a 2b serial-mantissa input (M_{IN}). ESICU, which contains an align-signal generator (ASG) and 16-lightweight converters (LCs), is designed to achieve a local 2b-serial alignment for G_{IN} depending on the exponent data read from the SRAM array. This approach eliminates the need for large-area subtractors and barrel shifters, compared to typical digital-alignment methods [1,2,5]. To fully utilize small-absolute-value negative-weight sparsity, N2CMAC computation is proposed to enable weight-mapping using a sign-magnitude data format; where, a signed floating-point or integer MAC is divided into an unsigned MAC and a sign-compensation, performed in the TS-A and CA&Q. The CIM macro supports four modes: one conventional read and write mode, and 3-computation modes: BF16A, BF16B, and INT8. BF16A and BF16B modes differ based on the preserved bit-width of the aligned mantissa, 10b for BF16A and 8b for BF16B, which are sufficient to handle the target CNN and transformer networks.

Figure 14.3.3 illustrates the ESICU structure and the embedded adaptive 2b-serial alignment scheme. The ESICU is composed of an ASG and 16 LCs: the ASG consists of 16 exponent adders (EAs), a maximum-value finder (MVF), a shift-destination tracer (SDT), and 16 equality comparators (ECs); the LC consists of an adaptive-reset register chain (ARRC), a self-alignment controller (SAC), and an output select and invert unit (OSIU). After the generation of exponent sums by EAs, the SDT value is determined based on the maximum sum from MVF, and decreases by 1 each cycle. Each EC then compares the SDT value with the higher 8 bits of its corresponding exponent sum (SUM[8:1]) and generates shift control signals (SHIFT0 - 15) for the LCs. If SHIFT = 0, the ARRC is in the store state, sequentially storing the serial minimum values; else, SHIFT = 1, the ARRC is in the shift state, sequentially outputting the previously stored minimum values: shifting out 2b per clock cycle. Then, OSIU generates OUT[1:0] and stores it as local input L_{IN}[1:0] for N2CMAC operations. The proposed adaptive 2b-serial alignment scheme reduces area overhead by 36.23%, compared to a traditional subtractor and barrel-shifter alignment schemes [1-6].

Figure 14.3.4 illustrates the detailed implementation of the format-mixed N2CMAC computation, along with the memory-bank configuration and waveforms for the 3 computation modes. Unlike previous 2's-complement MAC operations [11-13], weights are stored in the TS-As using a sign-magnitude format. The format-mixed N2CMAC computes MAC results for 2's-complement inputs and sign-magnitude weights, resulting in a 2's complement output. Computation consists of 4 steps: (1) the hierarchical input buffer provides G_{IN}, including its 8b parallel exponent in BF16A/B modes and a 2b serial M_{IN} in BF16A/B and INT8 modes; (2) the LC serially converts G_{IN} to L_{IN}, performing serial alignment in BF16A/B modes, and input-sign invert in BF16A/B and INT8 modes ({~L_{IN}[m-1], L_{IN}[m-2:0]}); (3) the digital multiply unit (DMU) inverts L_{IN} based on the sign of mantissa weight (M_W), multiplies L_{IN} by the magnitude of M_W, and the products are accumulated in an adder tree (AT); (4) the CA&Q shifts and accumulates the AT output over multiple cycles to generate PMACV, which is calibrated by adding a pre-set compensation value to the accumulator to obtain a MACV result: 20b result for INT8, 23b for BF16A, 21b for BF16B. The proposed format-mixed N2CMAC computation divides a signed MAC operation into an unsigned MAC and a sign-compensation. The unsigned MAC eliminates the overhead of sign-bit computation in the DMU and AT, while the sign-compensation value can be obtained either through two additional cycles or pre-set offline. This value is then shared across multiple cycles, further optimizing efficiency. In this work, memory-bank configuration is achieved through the MUX shared by adjacent columns. In BF16A/B modes, a staggered-mapping scheme is used for exponents and mantissa, while in INT8 mode the column-select (CS) signal functions as a column decoder, increasing the storage-compute ratio (SCR).

Figure 14.3.5 highlights the proposed schemes' performance. The serial-alignment and serial-MAC schemes proposed in this work achieve a 3.83× and 1.56× reduction in area and a 1.71× and 1.20× reduction in power, compared to prior parallel-alignment schemes using a parallel MAC [6] and a parallel-alignment scheme using a serial MAC [1,2]. The proposed N2CMAC computation reduces the required number of computing cycles for different networks with varying input precisions: a 19.80 and 16.50% reduction for 8b and 10b inputs using ViT (DeiT-S) @ImageNet. The use of sign-magnitude formatted weights significantly improves the sparsity of negative INT8 weights: showing a 1.83× bit-level sparsity increase for ResNet50 compared to a 2's complement format. The format-mixed N2CMAC circuit achieves a 1.96, 1.25, and 1.18× lower power consumption, latency, and area overhead compared to typical 2's complement MAC circuits [1,2].

Figure 14.3.6 presents measured results for the test chip, fabricated using a 28nm CMOS technology, implementing a 64kb B-A-N2CMAC FP-CIM macro. For BF16A/B-mode MAC operations, the measured access time is 5.4ns at 0.9V for BF16A/B inputs and weights and FP32 outputs. The maximum energy and area efficiency are 62.84TFLOPS/W and 697.17GFLOPS/mm^2 for BF16A/B MAC operations from a 0.55 - 0.9V supply when running ResNet50 @ImageNet. Compared to prior SRAM CIMs [1-5], this work improves the IN × W × memory density × norm. energy efficiency × output ratio FoM by 1.93 - 116.90×. This work achieves a 79.818% inference accuracy when used on ViT @ImageNet. Figure 14.3.7 shows the die photograph and chip performance summary table. All input/weight sparsity and toggle rates are bit-level and all simulation and measurements are obtained at room temperature (300K).

Acknowledgement:
This work was supported in part by the National Science and Technology Major Project under grant 2022ZD0118902; in part by the National Natural Science Foundation of China under grants 6506009793, 92264203 and 62204036; in part by the Key Research and Development Program of Jiangsu Province under grant BE2023020-1; in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under grant KYCX23_0344; and, in part by the Fundamental Research Funds for the Central Universities under grant 2242022k60009. The corresponding author is Xin Si (xinsi@seu.edu.cn).

 979-8-3315-4101-9/25/$31.00

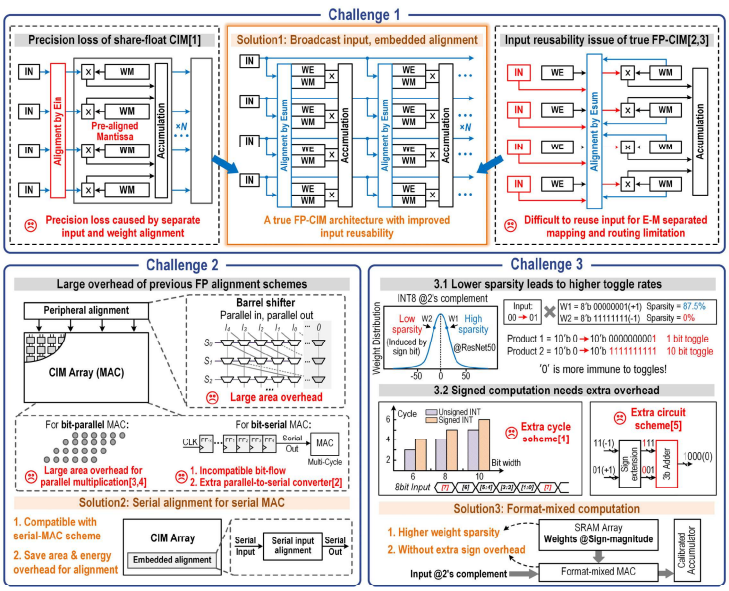
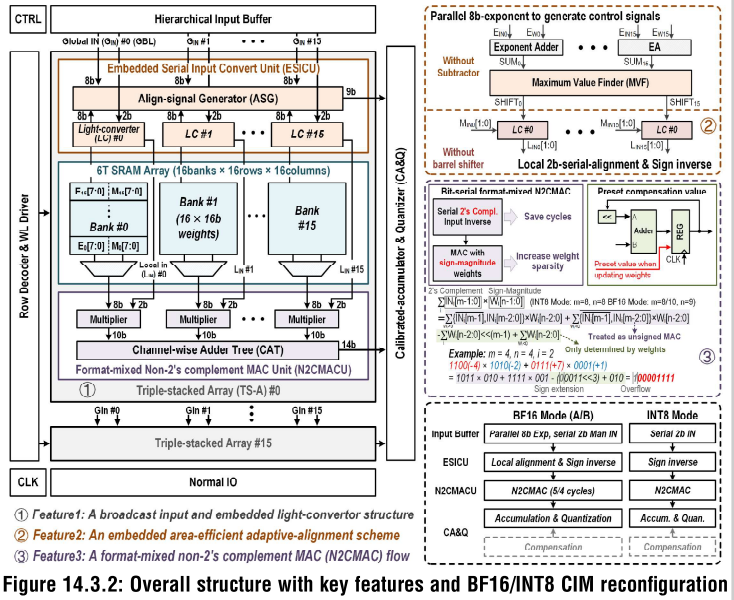
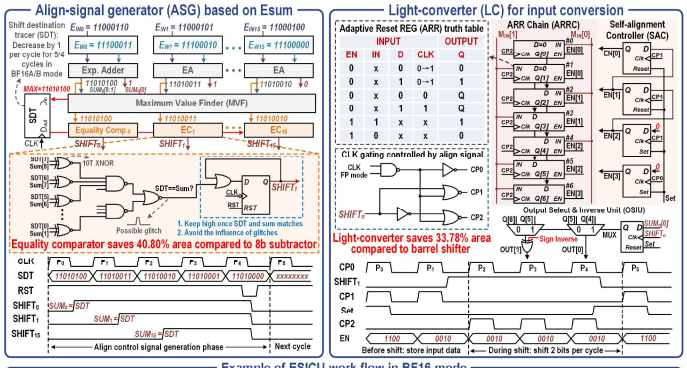


Figure 14.3.1: FP-CIM design challenges and proposed solutions.



① Feature1: A broadcast input and embedded light-convertor structure
② Feature2: An embedded area-efficient adaptive-alignment scheme
③ Feature3: A format-mixed non-2's complement MAC (N2CMAC) flow

Figure 14.3.2: Overall structure with key features and BF16/INT8 CIM reconfiguration of the proposed B-A-N2CMAC FP-CIM macro.



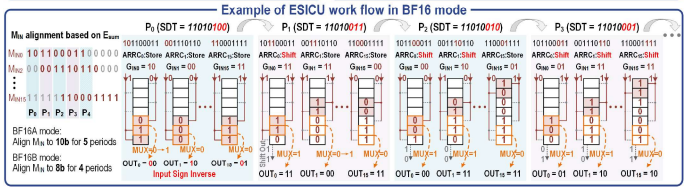
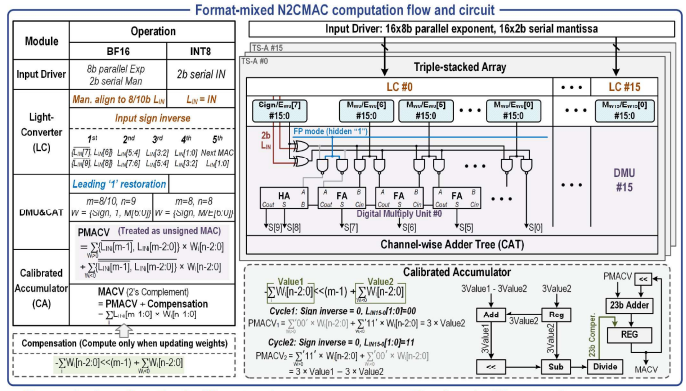


Figure 14.3.3: The ESICU structure for the embedded 2b-serial alignment, ASG for control signal generation, LC for input conversion (top). An example of ESICU workflow in BF16 mode (bottom).



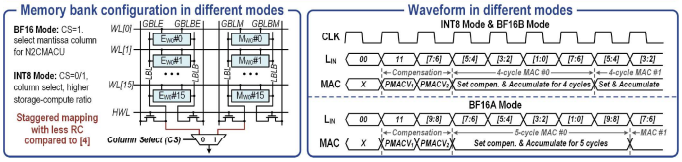
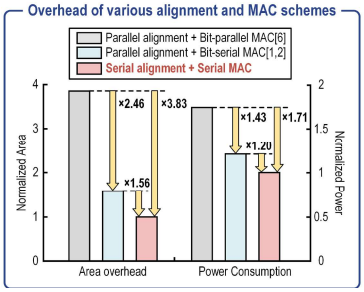
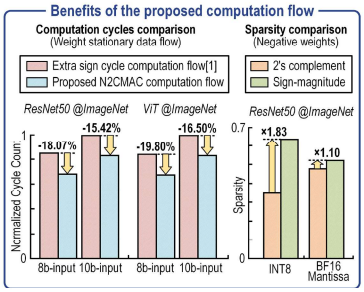
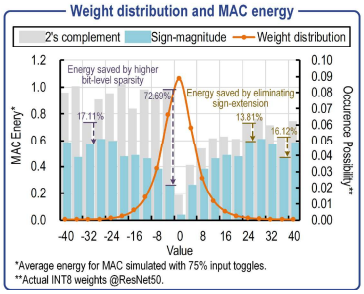


Figure 14.3.4: Detailed implementation of format-mixed N2CMAC computation (top). Memory bank configuration and MAC operation waveforms for 3 computation modes (bottom).







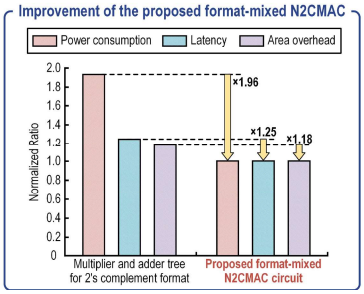
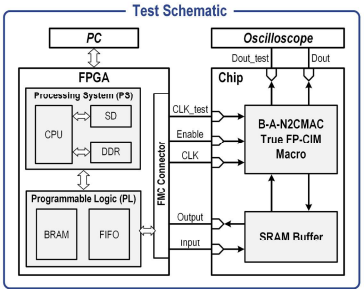
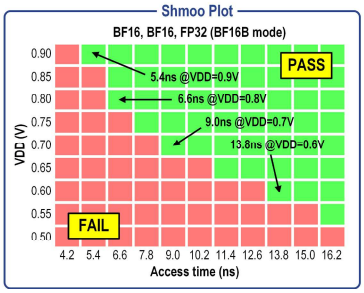
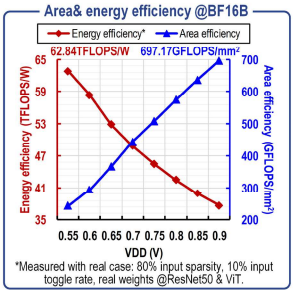


Figure 14.3.5: Proposed CIM macro simulated performance.







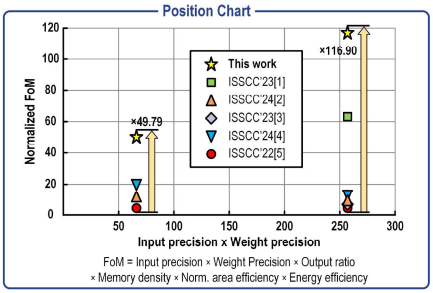
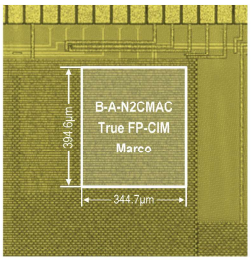


Figure 14.3.6: Measurement results and FoM comparison to prior work.



CHIP SUMMARY		
Technology	28nm CMOS	
Cell structure	6T+ESICU&N2CMAC	
Macro size	64Kb	
Macro area (mm^2)	0.136	
Input precision (bit)	8	BF16
Weight precision (bit)	8	BF16
Output precision (bit)	20	FP32
Supply voltage (V)	0.55-0.9	
Output ratio	1	
Access time (ns)	4.8	5.4/6.8
Memory Density (Kb/mm^2)	470.59	
Energy efficiency (TOPS/W) (TFLOPS/W)	[1]25.47-[2]90.15	[1]17.83-[2]62.84
Inference accuracy (%) ([3]ResNet50 @ImageNet)	76.030	76.130/76.130
Inference accuracy (%) ([4]ViT @ImageNet)	-	79.806/79.818
Inference mAP50 (%) ([5]YOLOv8 @COCO)	56.968	57.970/57.973

[1]Measured with worst case under 0.9V. Worst case: 25% input sparsity, 50% input toggle rate, 15% weight sparsity.
[2]Measured with real case under 0.55V. Real case: 80% input sparsity, 10% input toggle rate, real weights @ResNet50 & ViT.
[3]Software baseline is 76.140%.
[4]Data-efficient Image Transformer-Small (DeiT-S), software baseline is 79.834%.
[5]Software baseline is 57.974%.

Figure 14.3.7: Die micrograph and chip performance and key-metric summary table.

References:
[1] 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," *ISSCC*, pp. 128-130, 2023.
[2] W.-S. Khwa et al., "A 16nm 96Kb Integer/Floating-Point Dual-Mode-Gain-Cell-Computing-in-Memory Macro Achieving 73.3-163.3TOPS/W and 33.2-91.2TFLOPS/W for AI-Edge Devices," *ISSCC*, pp. 568-570, 2024.
[3] 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," *ISSCC*, pp. 126-128, 2023.
[4] Y. Yuan et al., "A 28nm 72.12TFLOPS/W Hybrid-Domain Outer-Product Based Floating-Point SRAM Computing-in-Memory Macro with Logarithm Bit-Width Residual ADC," *ISSCC*, pp. 576-578, 2024.
[5] F. Tu et al., "A 28nm 29.2TFLOPS/W BF16 and 36.5TOPS/W INT8 Reconfigurable Digital CIM Processor with Unified FP/INT Pipeline and Bitwise In-Memory Booth Multiplication for Cloud Deep Learning Acceleration," *ISSCC*, pp. 254-256, 2022.
[6] J. Saikia et al., "FP-IMC: A 28nm All-Digital Configurable Floating-Point In-Memory Computing Macro," *European Solid State Cir. Conf.*, pp. 405-408, 2023.
[7] Y. Wang et al., "A 28nm 83.23TFLOPS/W POSIT-Based Compute-in-Memory Macro for High-Accuracy AI Applications," *ISSCC*, pp. 566-568, 2024.
[8] H. Fujiwara et al., "A 3nm, 32.5TOPS/W, 55.0TOPS/mm2 and 3.78Mb/mm2 Fully-Digital Compute-in-Memory Macro Supporting INT12 × INT12 with a Parallel-MAC Architecture and Foundry 6T-SRAM Bit Cell," *ISSCC*, pp. 572-574, 2024.
[9] Y. He et al., "A 28nm 2.4Mb/mm2 6.9-16.3TOPS/mm2 eDRAM-LUT-Based Digital-Computing-in-Memory Macro with In-Memory Encoding and Refreshing," *ISSCC*, pp. 578-580, 2024.
[10] H. Mori et al., "A 4nm 6163-TOPS/W/b 4790–TOPS/mm2/b SRAM Based Digital-Computing-in-Memory Macro Supporting Bit-Width Flexibility and Simultaneous MAC and Weight Update," *ISSCC*, pp. 132-134, 2023.
[11] B. Wang et al., "A 28nm Horizontal-Weight-Shift and Vertical-feature-Shift-Based Separate-WL 6T-SRAM Computation-in-Memory Unit-Macro for Edge Depthwise Neural-Networks," *ISSCC*, pp. 134-136, 2023.
[12] A. Guo et al., "A 22nm 64kb Lightning-Like Hybrid Computing-in-Memory Macro with a Compressed Adder Tree and Analog-Storage Quantizers for Transformer and CNNs," *ISSCC*, pp. 570-572, 2024.
[13] T.-H. Wen et al., "Fusion of memristor and digital compute-in-memory processing for energy-efficient edge computing," *Science vol.* 384, 325-332, 2024.

979-8-3315-4101-9/25/$31.00 ©2025 IEEE

14.4 A 51.6TFLOPs/W Full-Datapath CIM Macro Approaching Sparsity Bound and <2^{-30} Loss for Compound AI

Zhiheng Yue*, Xujiang Xiang*, Yang Wang, Ruiqi Guo, Huiming Han, Shaojun Wei, Yang Hu, Shouyi Yin

Tsinghua University, Beijing, China

*Equally Credited Authors (ECAs)

Large-language models (LLM) are widely used with exceptional performance, but their prohibitive size and cost limits deployment on edge devices. The compound-AI combines several specialized small models to achieve matched or even superior accuracy on target downstream tasks [1,2]: e.g., RevCol [3] fuses multiple convolutional models and surpasses the 7.2B-parameter monolithic LLM model [4] on the ImageNet classification task by 1.64%. Therefore, the shift to compound systems opens opportunities for edge deployment in addition to model-size scaling. SRAM-based floating point (FP) CIM promises accelerated edge-AI models [5-14]. Figure 14.4.1 shows a conventional FP CIM macro, which faces three challenges for compound-AI acceleration: (1) Previous FP CIMs rely on a Gaussian data distribution to reduce alignment loss, where data is distributed around its mean [11]. However, compound-AI employs diverse models, with multiple data distributions, leading to an accuracy degradation of more than 5% [15]. (2) In variable compound-AIs, the CIM only performs MAC operations [16-23]; however, partial-sum/residual accumulation requires data movement and accounts for 26-72% power consumption, especially for bit-serial CIMs. (3) Conventional CIMs explore sparsity according to specific model features [11,12], but compound-AI models have flexible sparse patterns, causing a gap between the actual and theoretical speedup (1/1-sparsity).

We address these challenges by implementing an FP-CIM macro with three key features: (1) A heterogeneous-CIM macro, which supports parallel mantissa multiplication and exponent accumulation, allows post-product alignment to approach < 2^{-30} accuracy loss. The proposed mantissa^{EE} format supports different data precision for lossless computing. (2) The full data path is computed in memory to minimize data-movement power, and a hybrid format is explored to achieve peak-energy efficiency. (3) A network-in-memory (NIM) and a dynamic launcher capture static and dynamic sparsity opportunities, achieving >2.7× performance improvement compared to prior work. Figure 14.4.2 presents the overall architecture of the proposed macro, which comprises of 16 heterogeneous-CIM cores to deploy specialized models. Each core includes 8 CIM-sets and an inter-set adder. A 0.5kb accumulate-in-memory is used for partial-sum and residual accumulation. The CIM-set performs 4 parallel FP16 or 8 FP8 MACs within a 1.375kb mantissa-exponent CIM (ME-CIM), also included is a 0.25kb NIM and dynamic launcher for sparsity-aware acceleration. This CIM macro is designed as an edge accelerator to accommodate representative compound-AI task models and improves the accuracy × power × performance FoM.

While conventional CIM macros adopt rail-to-rail logic for FP computing, the mantissa inputs have already been deteriorated, due to alignment shifting, especially when the data is widely distributed with diverse exponent values [5-6,8-9]. Figure 14.4.3 presents the proposed workflow, in which the mantissa and exponent are first computed in parallel in the CIM and the alignment is postponed when the CIM generates partial products. Therefore, the CIM-computing error is mitigated and only occur during CIM-product accumulation. A proof-of-concept illustrates that the maximum FP16-MAC error rate is lower than 2^{-30}, compared to 2^{-1} in an earlier FP macro [5][12]. This lossless CIM design adapts to different data precision with Mantissa^{EE} transforms. Specifically, the unpacked mantissa of FP16 is shifted by the least significant 2b of the exponent to form a 16b Mantissa^{EE}, which is compatible with FP8 and INT8 representations, and reduces the required exponent width for alignment. The Mantissa^{EE} and exponent are stored in the same row of the ME-CIM, and perform multiply/add with inputs in parallel. The array integrates the XOR/AND logic necessary for both a Booth multiplier and a full adder (FA). Based on the required data precision, the surrounding shifter/partial FA is activated to complete the remaining logic. Lossless CIM products are then accumulated after post-alignment, which shifts the mantissa according to the exponent difference. The Mantissa^{EE} format concentrates the exponent distribution, and the post-alignment further takes the benefits of a higher product width to reduce truncation loss. The signal-error ratio (SER) evaluates hardware accuracy: it is defined as $10\log_{10}(\text{signal}/\text{error})$ [24]. This macro demonstrates an average of 60.1dB SER across 64×64 samples, which is 3.14× higher than a conventional FP CIM. Applied to compound models, ResNet-series, MobileNet-series, Attention-mechanism, and others, this work lowers the accuracy loss by 11.07×, on average.

Figure 14.4.4 shows the accumulate-in-memory (AIM) macro, which reduces data movement by performing in-situ accumulation. The data is stored in a 6T cell and uses $N_1/N_2/N_3$ for decoupled read and accumulation. The stored content is added with the CIM produced partial sum or residual connection within a pseudo-row. First, XOR logic distinguishes between INT and FP modes. Two elements are used for accumulation in INT mode. FP mode requires an extra subtraction to compare two operands for alignment. When CWL and CIM_{EN} are activated, the stored data and input together decide the CBL/CBLB voltage; when the cell stores 1 and IN equals 1, only then the N_1 is cut-off and N_4 drives CBL to a high state after pre-charge; thus, realizing an AND operation. Symmetrically, CBLB performs a NOR operation. Next, a partial FA takes advantage of the produced logic to simplify the adder design, and the summation output is written back to the cell. In FP32 mode, the FA output is utilized for alignment, accumulation, and further rounding. Compared with a full-digital scheme, AIM saves 38.3% and 15.4% power in INT32 and FP32 modes. Deployed on variable models, AIM reduces system power by 22.1 and 9.7% in INT and FP modes.

The power consumption of the multiplication data path is optimized by leveraging the best combination of multiplier/multiplicand format. Prior energy efficient multipliers choose sign-magnitude (SM) in place of the 2's complement (2C) format, but these formats do not accommodate a CIM structure [25]. The Booth-multiplier in ME-CIM achieves bit-parallelism by encoding inputs and correspondingly toggling and shifting weights. Therefore, a hybrid 2C (IN) and SM (W) combination incurs the fewest bit toggles and avoids adding-1 compensation, which reduces toggle activity by 5.2-47.5%; thereby, reducing power by 6.3-42.7% compared to other formats.

Figure 14.4.5 shows the static-dynamic sparsity-aware accelerator used to adapt to flexible compound-model sparse patterns. The static router targets the stationary sparsity scenarios: e.g., weight pruning and attention masking. The most-widely accepted 2:4 compression is used to maintain compatibility across different models [26]. The NIM selects 2-input elements from each of the 4 cells, according to the matched weight. Two improbable route paths are noted and excluded to reduce network complexity. Readout data are then Booth encoded to perform weight multiplication. As Figure 14.4.5 illustrates, each term is represented using 4 fields: ID, sign, valid, and SF. The encoding terms contain random valid distributions: i.e. dynamic sparsity. A Booth pool-lane sharing (BPLS) mechanism relocates non-zero terms to approach the ideal speedup due to data sparsity: specifically, 8 or 4 parallel inputs each with 4 or 8 encoding terms form a Booth pool. The dynamic launcher collects valid Booth-pool fields and in parallel leading-1 detectors iterate to find valid terms in sequence. Valid terms are retained and applied to the CIM for selective Booth-encoding multiplication. The ID field selects matched weights from the array, XOR logic toggles the sign bit in the SM representation, AND logic masks any non-valid elements, and the SF field provides shifting offset. By combining static and dynamic sparsity techniques, this macro achieves a 5.30× speedup on average on compound models: a 2× static and 2.65× dynamic speedup. Performance approaches 92.6% of the sparsity speedup bound. To further explore sparsity power benefits, a no pre-charge dual-port bit cell is implemented. The bit cell uses an inverter to eliminate the power-consuming pre-charge circuit. Charging and discharging power is consumed only when the RBL voltage changes. Therefore, this cell naturally suits sparse-data applications as RBLs are likely to read out consecutive 0s. Compared to an 8T dual-port cell, the 10T cell improves power consumption by 1.95× with a 1.26× area increase [27].

Figure 14.4.6 shows the test-chip measurement results. The test chip is fabricated in 28nm and occupies a 2.67mm^2. The 224kb heterogeneous macro integrates MAC, routing and accumulating operations for an all-in-memory data path; thus, achieving a peak efficiency of 16.2TFLOPs/W, and an average efficiency of 5.55 - 12.54TFLOPs/W when deployed on compound models. A post-product alignment enhances the signal-to-error ratio to 60.1dB, accomplishing an average error rate below 0.6% of all models in Figure 14.4.3. The NIM and launcher together explore static/dynamic sparsity to boost performance to nearly the speedup bound. A pre-charge-free cell also helps reduce access power for sparse data scenarios: allowing the macro to achieve 51.6TFLOPs/W and 2.44TFLOPs/mm^2. With all techniques, the proposed macro achieves a SER × system throughput/area / power FoM that is >5.72× higher than prior FP-CIMs. Figure 14.4.7 presents the die photograph.

Acknowledgement:
This work is supported in part by the National Key R&D Project under grant 2023YFB4403100; in part by the NSFC under grant 62125403, and grant 92164301; in part by the National Science and Technology Major Project under grant 2022ZD0115201; in part by the National Key Research and Development Program under grant 2021ZD0114400; Beijing S&T Project Z221100007722023; in part by the Beijing National Research Center for Information Science and Technology; and, in part by the Beijing Advanced Innovation Center for Integrated Circuits. (Corresponding author is Shouyi Yin, yinsy@tsinghua.edu.cn)

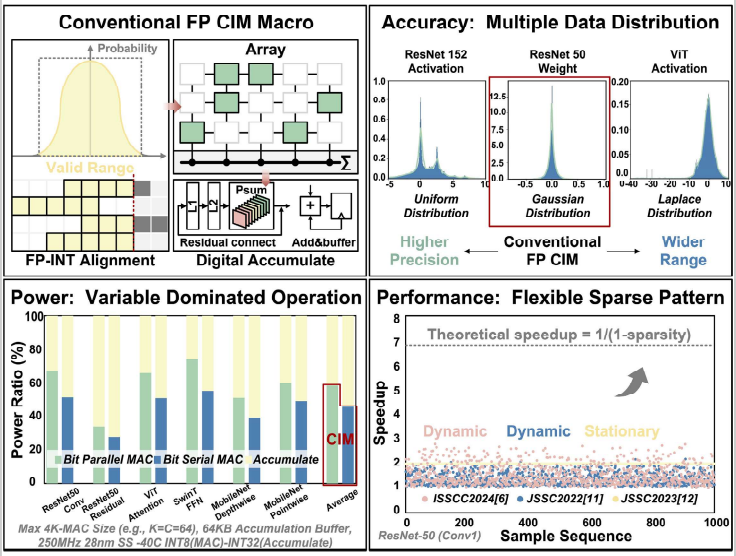


Figure 14.4.1: Challenges of deploying compound models on conventional FP CIM macro in 3 dimensions: accuracy, power, and performance.

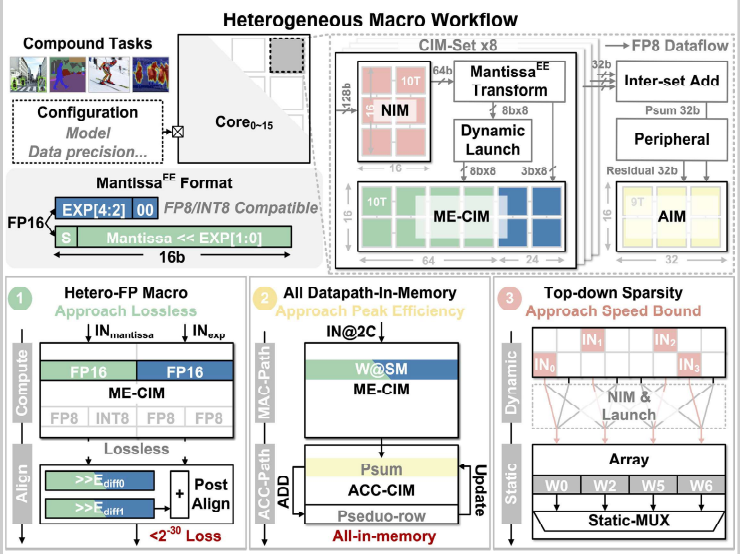


Figure 14.4.2: Overall architecture of heterogeneous macro and its three key features to approach peak accuracy, energy efficiency, speed.

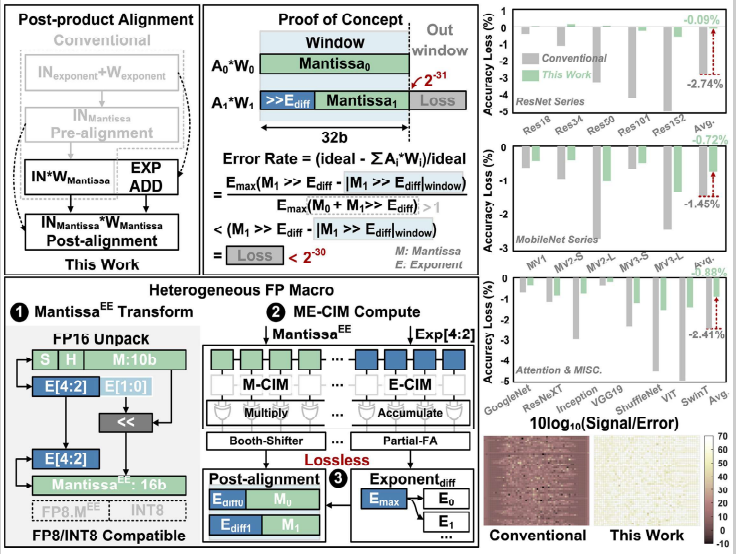


Figure 14.4.3: Post-CIM-alignment computing flow, error-rate proof of concept, and corresponding heterogenous CIM-macro design.

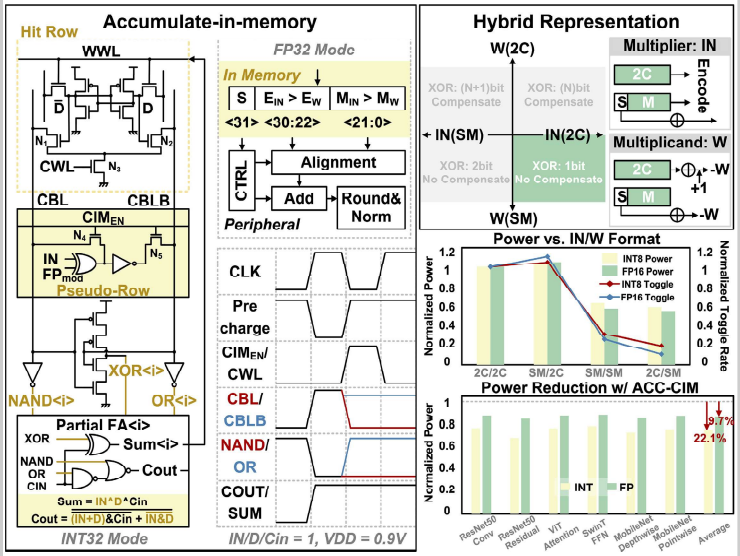


Figure 14.4.4: Accumulate-in-memory design and hybrid format multiplication optimization.

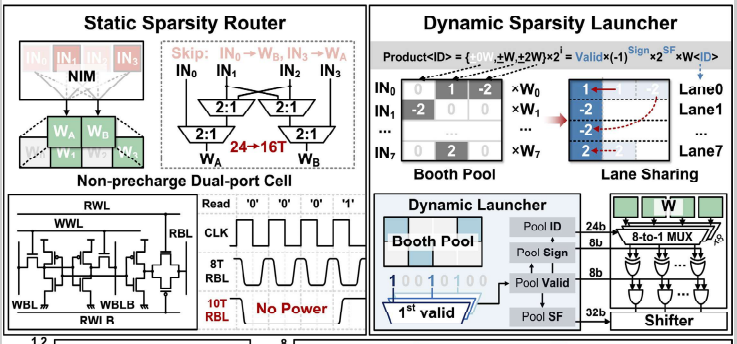
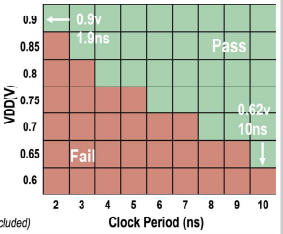


Figure 14.4.5: Static/dynamic sparsity-aware acceleration and corresponding NIM, sparse-aware cell, and dynamic launcher design.

Model	ResNet50 *1		MobileNetv2 *2		Vit_B *3
Dataset	ImageNetv2		ImageNetv2		ImageNetv2
Data Precision	INT8	FP16	INT8	FP16	FP16
Bit-level Sparsity	85.13%	71.08%	77.84%	66.74%	49.25%
Top-1 Accuracy	75.42%	76.30%	69.858%	70.25%	79.05%
Chip Accuracy Loss	-0.0%	-0.1%	-0.0%	-0.9/5%	-1.52%
Energy Efficiency (TFLOPS/W) *1	19.76	9.24	7.41	5.55	12.54
Throughput (GOPS) *2	344.8	114.1	73.5	38.1	154.9
Execute time (ms) *2	8.024	26.956	6.032	11.638	400.045
Inference/s *2	112	37	165	85	2

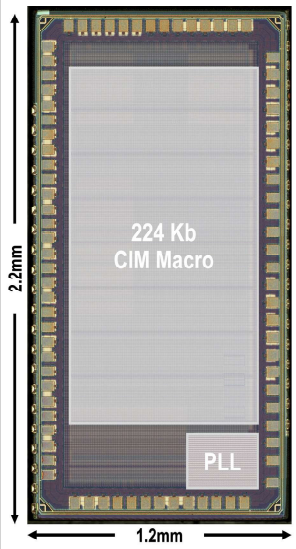
*1: @0.62V, 100MHz, full chip; *2: @0.9V, 500MHz; *3: ResNet50/MobileNetv2/Vit_B16: All Conv-layers/GEMM + Residual connection + Considering off-chip data movement (Non-linear layers are excluded)



	ISSCC2024[5]	ISSCC2024[6]	ISSCC2024[7]	ISSCC2023[8]	ISSCC2023[9]	ISSCC2023[10]	This Work
Technology	16nm	28nm	3nm	22nm	28nm	28nm	28nm
CIM Function	MAC	MAC	MAC	MAC	MAC	MAC	MAC/Route/ Accumulate
Voltage (V)	0.4~0.8	0.7~0.95	0.525~1.0	0.6~0.8	0.6~0.9	0.397~0.9	0.62~0.9
Frequency (MHz)	250	208	600~1800	156	147	10~400	100 ~ 525
Macro Size	96Kb	192Kb	156KB	832Kb	64Kb	16Kb	224Kb
Die Area (mm²)	8	1.94	1.37	18	0.146	4.536	2.67
IN/W Precision	BF16	BF16	INT12	BF16	BF16	FP16	FP16/FP8/INT8
Output Precision	FP32	8b ADC	31b	FP32	BF16	21b	FP32/INT32
Full Precision	Yes	No	Yes	Yes	No	No	Yes
Energy (TFLOPs/W) *1	33.2~45.4	13.55~32.78	16.3~23.2	16.22~17.59	14.04~31.6	3.2~16.9	15.1~51.0 (MAC) *2 4.8~16.2(Chip) *3
Area (TFLOPs/mm²) *1	2.63	2.21	12	0.069	2.05	0.36~2.1(INT4)	0.46~2.44

*1: 1 Op = FP16 x FP16 *2: Evaluating only CIM Macro for FP16 MAC operation. Peak energy efficiency evaluated at 0.62V, 100MHz; *3: Full chip evaluation;

Figure 14.4.6: Measurement results for representative models; Shmoo Plot; Key-feature comparison table to prior work.



Application		Compound-AI
Models		ResNet-series, MobileNet-series, Attention-mechanism, etc.
CIM Type		Heterogenous CIM
Data format		FP: Mantissa^{EE}
Alignment		Post-CIM alignment
CIM Size		224Kb
Die Area		2.67mm^2
Array Area		ACIM: 54.9x23.6um^2 MECIM:96.2x17.2um^2 NIM: 25.5x18.2um^2(logic rule)
Accuracy	Signal Error Rate	60.1 dB
	Model Accuracy Loss	Avg. 0.6% in compound models
Power	Accumulation Reduction	15.4%~38.3%
	Multiplication Reduction	6.3%~42.7%
Performance	Area Efficiency	FP16: 0.46~2.44 TFLOPs/mm^2 FP8: 0.62~3.30 TFLOPs/mm^2 INT8: 0.72~3.81 TOPs/mm^2
	Model Acceleration	Avg. 5.30x with dynamic/static sparsity
Energy Efficiency (Macro)	MAC	FP16: 15.1~51.6 TFLOPs/W FP8: 37.4~99.7 TFLOPs/W INT8: 43.2~115.0 TOPs/W
	Accumulate	FP32: 2.9~9.9 TFLOPs/W INT32: 4.4~15.2 TOPs/W
Energy Efficiency (Full Chip)		FP16: 4.8~16.2 TFLOPs/W FP8: 11.7~31.2 TFLOPs/W INT8: 14.2~49.6 TOPs/W

Figure 14.4.7: Die photo and hardware specification table.

References:

[1] R. Prabhakar et al., "SambaNova SN40L: Scaling the AI Memory Wall with Dataflow and Composition of Experts," *arXiv: 2405.07518*, 2024.
[2] M. Zaharia, et al., "The shift from models to compound ai systems," *URI: https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems*, 2024.
[3] Y. Cai, et al., "Reversible Column Networks," *ICLR*, 2023.
[4] C. Riquelme, et al., "Scaling Vision with Sparse Mixture of Experts," *NeurIPS 2021*, pp. 8583-8595.
[5] W.-S. Khwa et al., "A 16nm 96Kb Integer/Floating-Point Dual-Mode-Gain-Cell-Computing-in-Memory Macro Achieving 73.3-163.3TOPS/W and 33.2-91.2TFLOPS/W for AI-Edge Devices," *ISSCC*, pp. 568-570, 2024.
[6] Y. Yuan et al., "A 28nm 72.12TFLOPS/W Hybrid-Domain Outer-Product Based Floating-Point SRAM Computing-in-Memory Macro with Logarithm Bit-Width Residual ADC," *ISSCC*, pp. 576-578, 2024.
[7] M.-E. Shih et al., "NVE: A 3nm 23.2TOPS/W 12b-Digital-CIM-Based Neural Engine for High-Resolution Visual-Quality Enhancement on Smart Devices," *ISSCC*, pp. 360-362, 2024.
[8] 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," *ISSCC*, pp. 126-128, 2023.
[9] 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," *ISSCC*, pp. 128-130, 2023.
[10] J. Yue et al., "A 28nm 16.9-300TOPS/W Computing-in-Memory Processor Supporting Floating-Point NN Inference/Training with Intensive-CIM Sparse-Digital Architecture," *ISSCC*, pp. 252-254, 2023.
[11] 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 JSSC*, vol. 57, no. 8, pp. 2560-2573, Aug. 2022.
[12] F. Tu et al., "ReDCIM: Reconfigurable Digital Computing- In -Memory Processor with Unified FP/INT Pipeline for Cloud AI Acceleration," *IEEE JSSC*, vol. 58, no. 1, pp. 243-255, Jan. 2023.
[13] W, Tai-Hao, et al. "A 22nm 16Mb Floating-Point ReRAM Compute-in-Memory Macro with 31.2 TFLOPS/W for AI Edge Devices," *ISSCC*, pp. 580-582, 2024.
[14] R. Guo et al., "A 28nm 74.34TFLOPS/W BF16 Heterogenous CIM-Based Accelerator Exploiting Denoising-Similarity for Diffusion Models," *ISSCC*, pp. 362-364, 2024.
[15] C. Guo et al., "ANT: Exploiting Adaptive Numerical Data Type for Low-bit Deep Neural Network Quantization," *MICRO*, pp. 1414-1433, 2022.
[16] G. Desoli et al., "A 40-310TOPS/W SRAM-Based All-Digital Up to 4b In-Memory Computing Multi-Tiled NN Accelerator in FD-SOI 18nm for Deep-Learning Edge Applications," *ISSCC*, pp. 260-262, 2023.
[17] P. Chen et al., "A 22nm Delta-Sigma Computing-In-Memory ($\Delta\sum$CIM) SRAM Macro with Near-Zero-Mean Outputs and LSB-First ADCs Achieving 21.38TOPS/W for 8b-MAC Edge AI Processing," *ISSCC*, pp. 140-142, 2023.
[18] K. Ueyoshi et al., "DIANA: An End-to-End Energy-Efficient Digital and ANAlog Hybrid Neural Network SoC," *ISSCC*, pp. 256-258, 2022.
[19] H. Fujiwara et al., "A 3nm, 32.5TOPS/W, 55.0TOPS/mm2 and 3.78Mb/mm2 Fully-Digital Compute-in-Memory Macro Supporting INT12 × INT12 with a Parallel-MAC Architecture and Foundry 6T-SRAM Bit Cell," *ISSCC*, pp. 572-574, 2024.
[20] Z. Jiang, S. Yin, J. -S. Seo and M. Seok, "C3SRAM: In-Memory-Computing SRAM Macro Based on Capacitive-Coupling Computing," in IEEE SSCL, vol. 2, no. 9, pp. 131-134, Sept. 2019.
[21] H. Jia, H. Valavi, Y. Tang, J. Zhang and N. Verma, "A Programmable Heterogeneous Microprocessor Based on Bit-Scalable In-Memory Computing," in IEEE JSSC, vol. 55, no. 9, pp. 2609-2621, Sept. 2020.
[22] S. Liu et al., "A 28nm 53.8TOPS/W 8b Sparse Transformer Accelerator with In-Memory Butterfly Zero Skipper for Unstructured-Pruned NN and CIM-Based Local-Attention-Reusable Engine," *ISSCC*, pp. 250-252, 2023.
[23] H. Zhu et al., "COMB-MCM: Computing-on-Memory-Boundary NN Processor with Bipolar Bitwise Sparsity Optimization for Scalable Multi-Chiplet-Module Edge Machine Learning," *ISSCC*, pp. 250-252, 2022.
[24] S. Kim et al., "Scaling-CIM: An eDRAM-based In-Memory-Computing Accelerator with Dynamic-Scaling ADC for SQNR-Boosting and Layer-wise Adaptive Bit-Truncation," *IEEE Symp. VLSI Circuits, 2023.*.
[25] H. An et al., "An 8.09TOPS/W Neural Engine Leveraging Bit-Sparsified Sign-Magnitude Multiplications and Dual Adder Trees," *ISSCC*, pp. 422-424, 2023.
[26] J. Choquette, E. Lee, R. Krashinsky, V. Balan and B. Khailany, "The A100 Datacenter GPU and Ampere Architecture," *ISSCC*, pp. 48-50, 2021.
[27] Hiroki Noguchi et al., "Which is the best dual-port SRAM in 45-nm process technology? — 8T, 10T single end, and 10T differential —" *ICICDT*, pp. 55-58, 2008.

 979-8-3315-4101-9/25/$31.00

14.5 A 28nm 192.3TFLOPS/W Accurate/Approximate Dual-Mode-Transpose Digital 6T-SRAM CIM Macro for Floating-Point Edge Training and Inference

Yiyang Yuan[1,2], Bingxin Zhang[1,2], Yiming Yang[3], Yishan Luo[1,2], Qirui Chen[3], Shidong Lv[3], Hao Wu[1,2], Cailian Ma[1,2], Ming Li[1,2], Jinshan Yue[1], Xinghua Wang[3], Guozhong Xing[1], Pui-In Mak[4], Xiaoran Li[3], Feng Zhang[1]

[1]Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Beijing Institute of Technology, Beijing, China, [4]University of Macau, Macau, China

SRAM CIM macros have been developed to enhance energy efficiency (EF) in edge-AI applications. However, most research has predominantly focused on inference [1-6], with relatively little exploration into the training process. Fine-tuning, during training, is critical for improving the accuracy of neural network models, which directly impacts the user experience. Unlike remote- or cloud-based training, on-device training offers advantages such as real-time response, power saving, and user-privacy protection. The training process primarily involves two phases: feed-forward (FF) and backpropagation (BP). The FF phase resembles the inference process, while BP phase requires the multiplication of the error gradient with the transposed weight matrix. Although several studies have introduced transpose CIM (T-CIM) to support both FF and BP [7-9], several challenges remain: (1) Previous work uses separate circuits for FF and BP, thereby diminishing area and energy efficiency due to lack off multiply-accumulate (MAC) circuit reuse; (2) Prior work is limited to integer (INT) formats; research indicates that INT representation can significantly reduce accuracy during training due to its lower resolution [10]. Fortunately, developments in pre-aligned FP-CIM schemes have been made [1,11,12], but these still suffer from accuracy losses due to mantissa truncation during the pre-alignment process. (3) Reliance on analog-CIM schemes leads to accuracy losses due to process, voltage, and temperature (PVT) variations, which further degrade training accuracy. Although digital CIMs (DCIM) can mitigate these issues, optimizing the tradeoffs between SRAM arrays and MAC circuits to simultaneously achieve high-memory density (MD) and area efficiency (AF) remains challenging [1]. Increasing the number of SRAM sub-bank rows and employing bit-parallel techniques can improve MD and AF [2], while approximate computing can enhance access speed and EF [3,4], as illustrated in Fig. 14.5.1.

To address these challenges, we propose a fully-digital-transpose floating-point (FP) SRAM-CIM macro with three key features: (1) A cyclic-weight-mapping 6T-SRAM array (CWM-SRAM) that enables array transpose operations thus reusing MAC circuits for FF and BP; (2) A DCIM architecture with a signed fixed-point mantissa encoding (SFME) and a vector-wise pre-alignment (VWPA) that supports FP8, BF16, INT4, and INT8 formats to meet diverse accuracy and energy efficiency requirements during training and inference; (3) An accurate/approximate dual-mode bit-parallel MAC circuit (DMBP-MAC) that is compactly integrated within the CWM-SRAM to improve MD, access speed, AF, and EF.

Figure 14.5.2 illustrates the overall architecture of the macro, consisting of a read/write control circuit, an activation rotate-and-align circuit, and two sets of CWM-SRAM and DMBP-MAC that are organized into an upper and a lower block. The read/write-control circuit provides address decoding and read/write signals to CWM-SRAM; the activation rotate-and-align circuit aligns the elements of the activation vector with those of the weight vector, performing SFME and VWPA; the CWM-SRAM is a 1-port 4b/cell 64×64 6T-SRAM designed to support different access patterns for FF and BP. The upper and lower blocks operate separately in 4b-MAC INT4 mode, while in INT8 mode each block processes 4 MSBs and 4 LSBs. In FP8 and BF16 modes the weight and activation mantissas are pre-aligned and encoded into 4 and 8b forms; thereby, reusing the INT4 and INT8 data path. DMBP-MAC consists of 64 accurate/approximate dual-mode bit-parallel multipliers (DM-MUL) and 6 levels of dual-mode adders (DM-ADD) to generate a 14b dot product. To facilitate system integration, the macro is designed to be compatible with the standard AMBA bus interface. The AHB interface handles memory accesses, while all computing-related control signals and MAC results are transmitted via APB.

Figure 14.5.3 illustrates the scheme of CWM-SRAM. The weight matrix is stored in CWM-SRAM using a cyclic scheme, where each row of the matrix shifts its elements to the right based on its row index. During the FF phase, weights are retrieved from each row. During the BP phase, the diagonal of the weight matrix is retrieved, which corresponds to the weight matrix's transposed form. This allows for both FF and BP weights to share the 256b read port connected to the MAC circuits, simplifying the SRAM access requirements to a single-port and enabling the reuse of MAC circuits. Since the original weight vectors are shifted, due to the cyclic scheme, the activation vectors also need to be right-shifted to maintain weight-vector alignment; this adjustment is performed in the activation rotator. To implement this scheme, we propose a structure that accesses every 4b cell independently. The activation of each 4b group is controlled by separate FF and BP read-enable signals. The FF read-enable signal connects 4-row cells and the BP read-enable signal connects 4-diagonal cells. To mitigate routing congestion, we implement the diagonal connection using the 3rd-metal layer and connect the corresponding diagonal in the 4th layer; the row connection is in the 5th layer. This method circumvents the need for numerous MAC circuits and read ports as is the case for previous T-CIM works [7,8], resulting in a reduction in area and power consumption.

Figure 14.5.4 depicts the DCIM architecture supporting FP8, BF16, INT4, and INT8 formats. FP8 is proposed as a novel FP data format to achieve higher training efficiency than BF16 [13]. FP8 can be represented in two distinct forms: E5M2 and E4M3. The E5M2 format, with a 5b exponent and a 2b mantissa, offers greater dynamic range but lower precision; the E4M3 format, with a 4b exponent and a 3b mantissa, provides a smaller dynamic range but higher precision. SFME is proposed to enable variable bit-width FP MACs. The FP weight $-1^{s}\cdot 2^{E-Bias}(1.M)$ is changed to $2^{E_{max}-Bias}[-1^{s}(0.M_n)]$ after pre-alignment, the original exponent changed to a shared exponent $E_s = E_{max} - \text{Bias}$ and the hidden bit together with the mantissa is normalized to M_n. Thus, the aligned mantissa can be represented as a signed complement fixed-point number: $M_a=-1^{s}\cdot 0.M_n$, with the decimal point placed behind the sign bit. The FP activation is also aligned before the FP-MAC operation. To support a unified MAC for 4 and 8b aligned mantissas, we designed a quad-mode 4b multiplier that covers all 4b partial-product combinations for 8b 2's complement multiplications. To reduce accuracy loss caused by pre-alignment mantissa truncation we used VWPA to perform pre-alignment for elements in the same vector, allowing us to benefit from the closer same-vector element distribution in the smaller truncation window.

Figure 14.5.5 illustrates the architecture of the DMBP MAC: comprising of a DM MUL based on the broken-array approximate multiplier (BAM) concept and an adder tree constructed using DM ADD that employs the lower-part OR-adder (LOA) principle [14]. The DMBP MAC operates in two modes, approximate or accurate, enabling an accuracy/efficiency trade-off depending on the application. Compared to a bit-serial approach, the bit-parallel approach offers higher speed, due to fewer compute cycles, and lower energy consumption, due to a reduced input toggle rate [2]. In approximate mode, the DM MUL discards the 6 partial sums on the right-hand side, and the DM ADD replaces the lower-part addition with a bitwise OR operation; a sparsity-aware bias adder is introduced into DM MUL to cancel the BAM offset error. Due to the shortened critical delay path and fewer toggled components, the approximate mode achieves a 12% faster computing speed and a 31% power reduction compared to the accurate mode. The mean relative-error distance (NMED) is the mean absolute value of error distance (the difference between the actual output and an ideal output) normalized to the maximum output value. To measure the effect of approximation on the computing result, 50,000 random input pairs are sampled, yielding a 5.3% NMED. The error distances exhibit a Gaussian distribution, with an average value close to zero. Additionally, a centrosymmetric layout is proposed to integrate the DMBP MAC with the CWM SRAM, optimizing the length-width ratio of macro and alleviating routing congestion.

As shown in Fig. 14.5.6, the 28nm-process FP-transpose DCIM macro achieves a measured access time of 2.5ns at 0.9V with a FP8 IN, FP8 W and BF16 OUT. The maximum FP EF is 192.3TFLOPS/W for FP8, measured at 0.55V under a 50% weight sparsity and a 50% input toggle rate in approximate mode. Compared to previous T-CIM work [7,8], this work shows significant improvements in EF and AF, attributed to the reuse of CWM-SRAM MAC circuits for FF and BP. By employing bit-parallel and approximate arithmetic in DMB -MAC, this work exhibits shorter access times and higher EF and AF than previous non-transpose DCIM work [1,5]. With the proposed VWPA, in accurate mode, this work shows lower accuracy loss compared with layer-wise pre-align FP CIM [1]. Using SFME, this work presents the first FP CIM capable of unifying four different INT/FP formats: INT4, INT8, FP8, and BF16. Measurements using ResNet-18@CIFAR10 and VGG-19@CIFAR100, FP8 show slightly lower inference accuracy compared to INT8 and BF16, but accuracy results significantly surpass INT4. In terms of EF, FP8 notably outperforms both INT8 and BF16, making it an optimal choice for balancing EF and accuracy. Figure 14.5.7 presents the die photo and summary table.

Acknowledgement:
This work was supported in part by National Key Research and Development Program of China 2023YFB4402400; in part by the National Natural Science Foundation of China under grant U2341218; in part by the Strategic Priority Research Program of the Chinese Academy of Sciences China under grants XDB44000000 and XDA330100; The corresponding author are Xiaoran Li (xiaoran.li@bit.edu.cn) and Feng Zhang (zhangfeng_ime@ime.ac.cn).

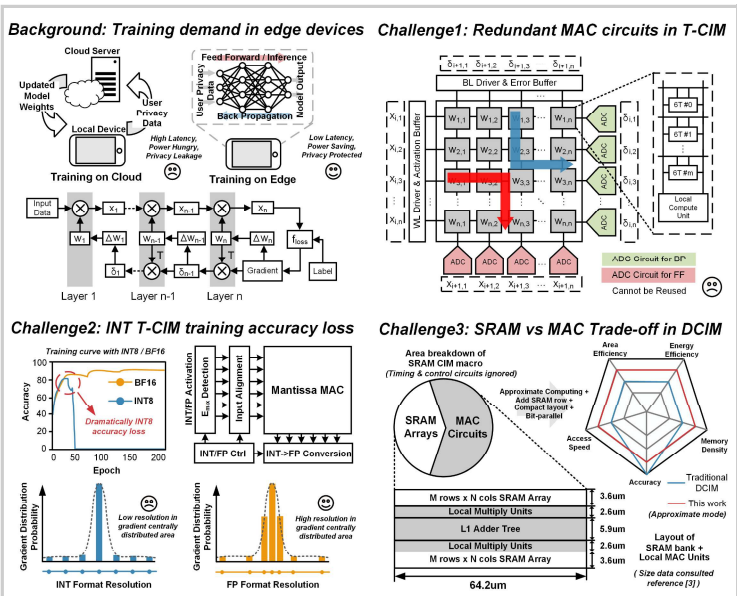


Figure 14.5.1: Challenges of transpose-CIM macro and motivation of this work.

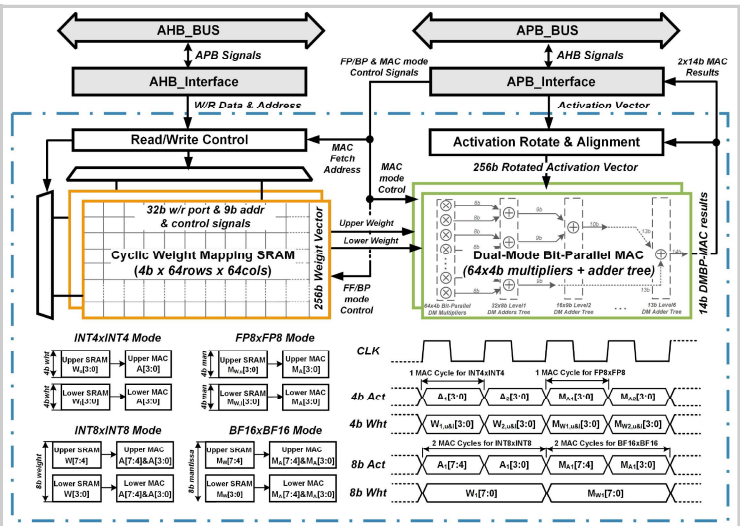


Figure 14.5.2: Overall architecture of the proposed transpose digital-SRAM CIM.

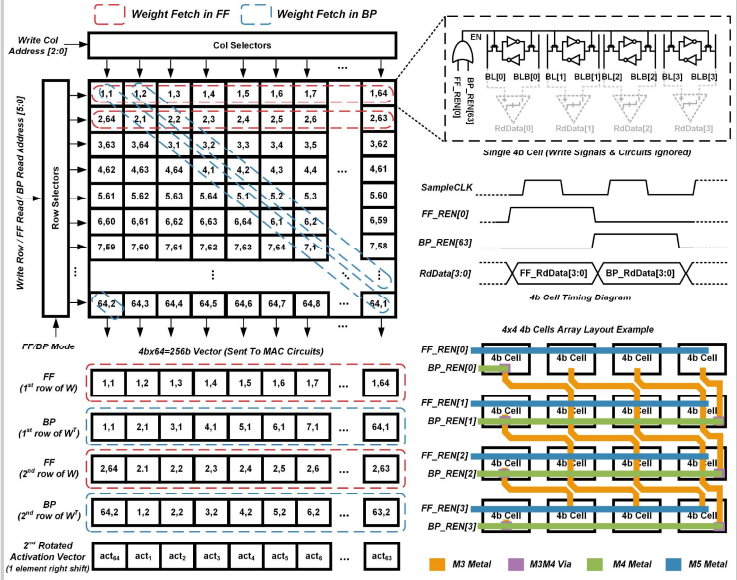


Figure 14.5.3: CWM-SRAM scheme with 6T bit cells for MAC circuit reuse.

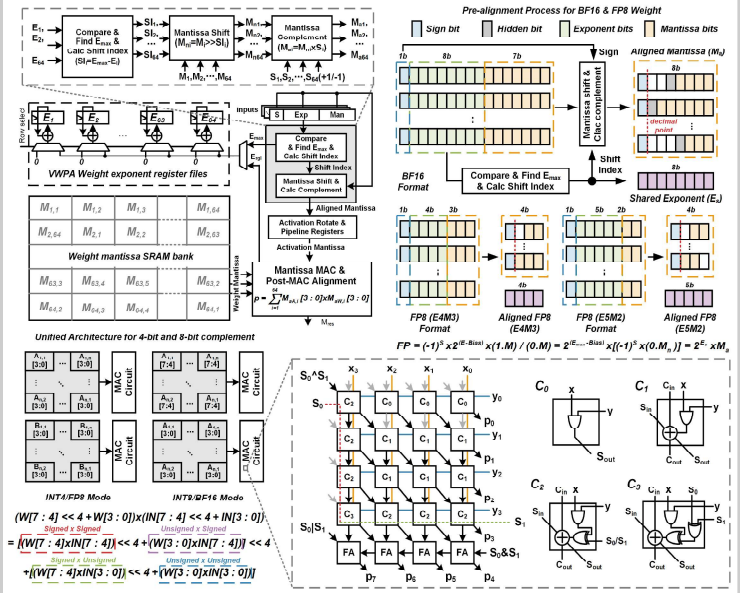


Figure 14.5.4: SFME scheme with quad-mode 4b multiplier and VWPA.

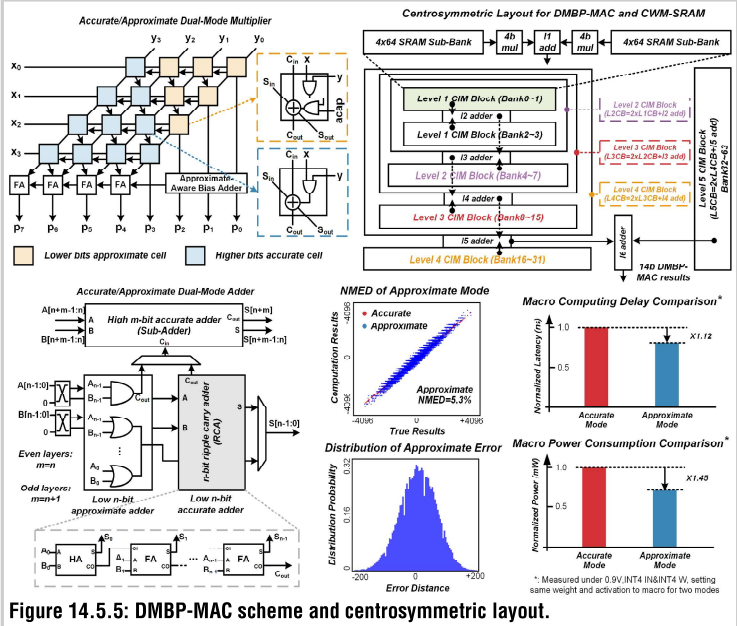


Figure 14.5.5: DMBP-MAC scheme and centrosymmetric layout.

	JSSC'22[7] J.-W. Su	ASSC'21[8] J.Song	ISSCC'23[11] A.Guo	ISSCC'24[5] Y.Wang	This Work Approximate Mode	This Work Accurate Mode
Technology	28nm	28nm	28nm	28nm	28nm	
CIM Type	Analog Transpose CIM	Analog Transpose CIM	Digital Non-transpose CIM	Digital Non-transpose CIM	Digital Transpose CIM	
Precision	INT2/4/8	INT1	INT8,BF16	POSIT8/16	INT4/8,FP8,BF16	
Access Time	15.2ns @INT8	---	8.8ns @BF16	---	2.5ns @FP8 5.1ns @BF16	2.7ns @FP8 5.4ns @BF16
Memory Density	---	0.273Mb/mm2	0.428Mb/mm2	---	0.732Mb/mm2	
Area Efficiency	---	0.056TOPS/mm2	2.57TOPS/mm2 2.05TFLOPS/mm2	2.74TFLOPS/mm2 1.35TFLOPS/mm2	3.23TOPS/mm2 2.34TFLOPS/mm2	2.89TOPS/mm2 2.24TFLOPS/mm2
INT Energy Efficiency	7-7.6TOPS/W @INT8	4TOPS/W @INT8	19.5-44TOPS/W @INT8	---	25.5-71.4TOPS/W @INT8	17.54-50TOPS/W @INT8
FP Energy Efficiency**	---	---	14.04-31.6TFLOPS/W @BF16	7.47-38.37TFLOPS/W @POSIT16	17.19-48.08 TFLOPS/W @BF16	12.78-36.44 TFLOPS/W @BF16
Accuracy Loss***	2.27% @CIFAR100	2.98% @CIFAR10	0.17% @CIFAR100	0.12% @ImageNet	0.05% @CIFAR10 0.29% @CIFAR100	0.01% @CIFAR10 0.1% @CIFAR100

* [8] take area efficiency for INT8, [1] INT8 and BF16, [5] POSIT8 and POSIT16, our work INT8 and BF16 ** All take average energy efficiency. [8] is normalized to INT8xINT8
*** [7] take accuracy for INT8xINT8, [8] take INT1xINT1, [1] take BF16xBF16, [5] take POSIT8xPOSIT8, our work take BF16xBF16

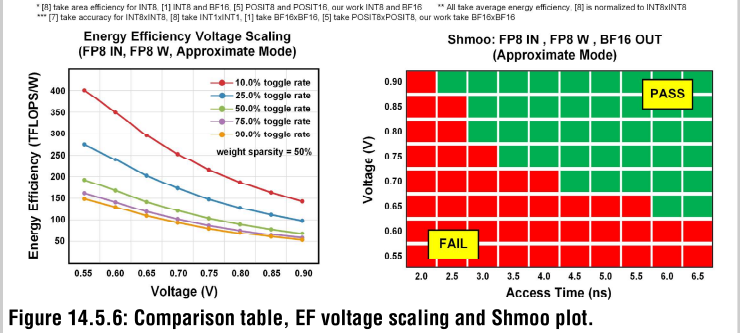
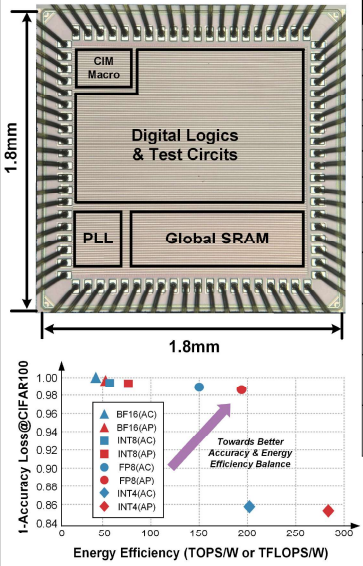


Figure 14.5.6: Comparison table, EF voltage scaling and Shmoo plot.



Chip Summary					
Technology		28nm			
Macro Area		0.043mm2 (253.8um x 168.3um)			
Macro Size		32Kb			
Supply Voltage		0.55-0.9V			
Number of Channels		128			
Input Precision		INT4	INT8	FP8	BF16
Weight Precision		INT4	INT8	FP8*****	BF16
Access Time	AP	0.9ns @0.9V	1.9ns @0.9V	2.5ns @0.9V	5.1ns @0.9V
	AC	1.0ns @0.9V	2.1ns @0.9V	2.7ns @0.9V	5.9ns @0.9V
Average Energy* Efficiency	AP	102-285.4 TOPS/W	25.5-71.4 TOPS/W	68.8-192.3 TFLOPS/W	17.2-48 TFLOPS/W
	AC	70.2-200 TOPS/W	17.5-50 TOPS/W	51.1-145.8 TFLOPS/W	12.8-36.4 TFLOPS/W
Peak Energy** Efficiency	AP	212.4-594 TOPS/W	53-148.5 TOPS/W	143-400 TFLOPS/W	35.8-100 TFLOPS/W
	AC	146.4-416.5 TOPS/W	36.5-104 TOPS/W	106.4-303.4 TFLOPS/W	26.6-75.8 TFLOPS/W
Accuracy Loss*** @CIFAR10	AP	3.70%	0.12%	0.42%	0.05%
	AC	3.62%	0.04%	0.40%	0.01%
Accuracy Loss**** @CIFAR100	AP	14.21%	0.41%	1.24%	0.29%
	AC	14.13%	0.30%	1.00%	0.10%

AP represents approximate mode, AC represents accurate mode
* Measured under 50% weight sparsity and 50% toggle rate
** Measured under 50% weight sparsity and 10% toggle rate
*** ResNet-18 model, TOP-1 accuracy, software baseline 95.47%
**** VGG-19 model, TOP-1 accuracy, software baseline 71.27%
***** Accuracy measured under FP8 E4M3

Figure 14.5.7: Die photo and summary table.

References:
[1] 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," *ISSCC*, pp. 128-130, 2023.
[2] H. Fujiwara et al., "A 3nm, 32.5TOPS/W, 55.0TOPS/mm2 and 3.78Mb/mm2 Fully-Digital Compute-in-Memory Macro Supporting INT12xINT12 with a Parallel-MAC Architecture and Foundry 6T-SRAM Bit Cell," *ISSCC*, pp. 572-574, 2024.
[3] Y. He et al., "A 28nm 38-to-102-TOPS/W 8b Multiply-Less Approximate Digital SRAM Compute-In-Memory Macro for Neural-Network Inference," *ISSCC*, pp. 130-132, 2023.
[4] D. Wang et al., "DIMC: 2219TOPS/W 2569F2/b Digital In-Memory Computing Macro in 28nm Based on Approximate Arithmetic Hardware," *ISSCC*, pp. 266-268, 2022.
[5] Y. Wang et al., "A 28nm 83.23TFLOPS/W POSIT-Based Compute-in-Memory Macro for High-Accuracy AI Applications," *ISSCC*, pp. 566-568, 2024.
[6] W.-S. Khwa et al., "A 16nm 96Kb Integer/Floating-Point Dual-Mode-Gain-Cell-Computing-in-Memory Macro Achieving 73.3-163.3TOPS/W and 33.2-91.2TFLOPS/W for AI-Edge
Devices," *ISSCC*, pp. 568-570, 2024.
[7] J.-W. Su et al., "Two-Way Transpose Multibit 6T SRAM Computing-in-Memory Macro for Inference-Training AI Edge Chips," *IEEE JSSC*, vol. 57, no. 2, pp. 609-624, Feb. 2022.
[8] J. Song et al., "A 16Kb Transpose 6T SRAM In-Memory-Computing Macro based on Robust Charge-Domain Computing," *A-SSCC*, pp. 1-3, Nov. 2021.
[9] H. Jiang et al., "CIMAT: A Compute-In-Memory Architecture for On-chip Training Based on Transpose SRAM Arrays," *IEEE Tran. on Comp.*, vol. 69, no. 7, pp. 944-954, Jul. 2020.
[10] S. Wang et al., "Gradient Distribution-Aware INT8 Training for Neural Networks," *Neurocomputing*, vol. 541, Article 126269, Jul. 2023.
[11] F. Tu et al., "A 28nm 29.2TFLOPS/W BF16 and 36.5TOPS/W INT8 Reconfigurable Digital CIM Processor with Unified FP/INT Pipeline and Bitwise In-Memory Booth Multiplication for Cloud Deep Learning Acceleration," *ISSCC*, pp. 255-257, 2022.
[12] T.-H. Wen et al., "A 22nm 16Mb Floating-Point ReRAM Compute-in-Memory Macro with 31.2TFLOPS/W for AI Edge Devices," *ISSCC*, pp. 580-582, 2024.
[13] P. Micikevicius et al., "FP8 Formats for Deep Learning," *arXiv:2209.05433*.
[14] H. R. Mahdiani et al., "Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications," *IEEE Tran. Cir. & Sys. I*, vol. 57, no. 4, pp. 850-862, Apr. 2010.

979-8-3315-4101-9/25/$31.00 ©2025 IEEE

14.6 A 28nm 64kb Bit-Rotated Hybrid-CIM Macro with an Embedded Sign-Bit-Processing Array and a Multi-Bit-Fusion Dual-Granularity Cooperative Quantizer

Xi Chen, Shaochen Li, Zhican Zhang, Wentao Zheng, Xiao Tan, Yuchen Tang, Yuhui Shi, Lizheng Ren, Yibo Mai, Feiran Liu, Jinwu Chen, Zhaoyang Zhang, An Guo, Tianzhu Xiong, Bo Wang, Xinning Liu, Weiwei Shan, Bo Liu, Hao Cai, Jun Yang, Xin Si

Southeast University, Nanjing, China

Hybrid-domain CIMs [1-3] are attracting increasing attention nowadays as they combine the advantages of both digital CIMs (DCIM) [4-6] and analog CIMs (ACIM) [7-10], offering a more balanced choice. As depicted in Figure 14.6.1, when designing a hybrid CIM (HCIM), the first challenge to address is (1) the definition of boundary between the digital and analog part, which is determined by feature input mode and weight mapping. Prior HCIMs either utilized a bit-parallel in features to obtain high accuracy at the cost of hardware (HW) overhead [1] or realized a low HW overhead but suffered from accuracy loss due to error pollution problem (error of the analog partial sum makes the high precision digital partial sum of the same bit weight meaningless) with bit-serial scheme and specific weight mapping strategy [2,5], neither of which provided a perfect solution to the boundary question. Therefore, we propose a bit-rotated feature-in scheme to address this challenge. To compress multiple multiplication results into one MAC result, several levels of shifters and adders are usually required, and the sign-bit needs to be considered when compressing signed and unsigned information. Unlike the bit-parallel/serial approach [1,2], the bit-rotated scheme considers the sign-bit in the first product-wise adder, which induces (2) more digital overhead for sign-bit processing. Another challenge that continues to plague bit-rotated HCIM is (3) excessive energy wasted on low accuracy-contributed low-bit quantization in analog parts.

The contributions of this work are as follows: (1) a hybrid structure with a bit-rotated feature-in scheme for low HW overhead and high accuracy; (2) an embedded sign-bit-processing (ESP) hybrid SRAM array for correct and low HW overhead computation; (3) a multi-bit-fusion dual-granularity cooperative quantizer (MF-DGCQ) for multi-bit multi-cycle low-power/delay quantization. The proposed 28nm 64kb bit-rotated hybrid-CIM macro achieves a high, 67.8TOPS/W, energy efficiency for INT8 MAC operations and represents the most balanced point of HCIM.

Figure 14.6.2 presents the overall structure and operations of the proposed bit-rotated hybrid-CIM macro. The overall structure consists of a 256 × 256b ESP hybrid SRAM array, a gather and rotate input buffer (GRIB), WL drivers, a controller (CTRL), write IO (WIO), an MFC-DGCQ, and a post processing unit (PPU). The ESP hybrid SRAM array consists of 16 channel-wise MAC units (CWMAUs). Each CWMAU includes 32 product-wise MAC units (PWMAUs) and a channel-wise adder (CWA). A sign-bit MAC cell (SBMC), 7 hybrid-MAC cells (HBMCs) and a product-wise adder (PWA) form a PWMAU. During GEMM, static weights are preloaded to the SRAM array, with the most significant bit (MSB) stored in the SBMC while other bits are in HBMCs. Features are fetched by the gather buffer (GB) and written to the array via GBL/GBLBs by the rotated-input buffers (RIUs) using the proposed bit-rotated method to perform product-wise first hybrid MAC operations, considering sign-bit processing in 8 cycles (take INT8 MAC as example). Digital MAC results, as well as the sign-bit extension, are accumulated by the PWA and CWA, while the analog part is computed in the charge domain; the output results are quantized by MFC-DGCQ. The two results are combined by the PPU. Thus, a vertical-cut hybrid pattern is achieved without using a bulky MAC array and digital adders.

Figure 14.6.3 illustrates detailed schematics and operations of the GRIB and ESP hybrid SRAM array. Each GB consists of 8 DFFs and is responsible for gathering serial input data and delivering them to the 8 corresponding RIUs in parallel. Each RIU comprises of two 2:1 MUXs and a DFF that work in fetch mode or rotate mode. A complete rotated-MAC operation occupies one fetch period and 7 rotate periods. During the fetch period (rot = 0), 8b features and 8 zeros are fetched simultaneously from the GB by the left and right DFFs of 8 RIUs. During the rotate period (rot = 1), features are transferred to adjacent DFFs within the left and right groups; particularly, feature data from the left DFF of RIU#7 will be transferred to the right DFF of RIU#0, and 0 will be transferred to the left DFF of RIU#0. There are two kinds of MAC cells in the ESP hybrid SRAM array: (1) the SBMC comprises of 16 6T-SRAM cells, a selective reverse digital multiplier (SRDM) and a sign-bit extension unit (SBEU); (2) the HBMC comprises of 16 6T-SRAM cells, an SRDM and an analog multiplier (AM). During each cycle, a 1b weight is read from one of the 16 6T-SRAM cells and multiplied by 2b features on GBL/GBLB; the result from the SRDM contributes to the digital part, while the one from the AM contributes to the analog part. The ESP hybrid SRAM array supports both unsigned and signed modes. When it works under signed mode, two aspects of sign-bit processing need to be considered: (1) selective reversion, where the SRDM would deliver the original multiplication results or the reversed ones, depending on whether F[7] is the multiplier or not; (2) sign-bit extension, where previous results (p07, p17, etc.) would be stored and reused by the 3b register and adder (3b_RA) in the SBEU. While in unsigned mode, the SRDM would deliver the original multiplication results and SBEU does not perform sign-bit extension.

Figure 14.6.4 shows the proposed MF-DGCQ for multi-bit multi-cycle low-power/delay quantization. In our proposed HCIM, an 8b ADC is utilized to quantize the analog MAC result; the accumulation length is set to 32 per cycle for each CWMAU. Conventional quantization schemes activate the ADC every cycle, resulting in a poor output ratio (OR: ratio of practical output resolution to the ideal output range) for low-bit quantization. The proposed multi-bit-fusion quantization scheme quantizes multiple low bits together to reduce the ADC activation time. The multi-bit-fusion circuit (MFC) comprises of 3 sets of write (WR) and read (RD) switches, 3 fusion capacitors and a compensation capacitor, whose capacitance ratio is 1:2:4:1. In the reset phase (period#0), all data on GBLB is 0, which means the upper plate of the computation capacitor in the PWMAU is connected to 0. Meanwhile, all switches in the MFC are closed, to connect the bottom plates of the computation capacitors, the upper plates of fusion capacitors, and the compensation capacitor to 0. In the write phase, coupling voltages V_c that represent MAC results are obtained on the fusion capacitors successively. In the read phase, corresponding RD switches are closed to perform charge sharing between the fusion and compensation capacitors; the obtained sharing voltage, V_s, represents the multi-bit MAC result. V_s is quantized by the DGCQ, and then reset to 0. The proposed DGCQ consists of a coarse-grained and a fine-grained part. The coarse-grained quantizer (CGQ) comprises a comparator, a 4b counter, and a voltage subtraction circuit (VSC). The fine-grained quantizer (FGQ) is an 8b SAR-ADC. When V_s from the MFC is ready, it is transported to the DGCQ and performed charge sharing with the prior accumulation voltage V_{acc}. In most quantization cycles (when the CGQ activated), the new V_{acc} is compared with V_{cm} ($^1/_4V_{DD}$) to determine the behavior of the counter and VSC. The counter records the times when $V_{acc} > V_{cm}$, and the VSC regulates V_{acc}. Once multi-cycle computation is done, the FGQ is activated to quantize the remaining V_{acc}. Both coarse-grained and fine-grained results are delivered to the PPU to obtain the final MAC result.

Figure 14.6.5 presents the simulation results of the proposed HCIM macro. We evaluate our proposed bit-rotated structure on compute error by performing a 10k Monte Carlo simulations. Compared to prior hybrid structures [1,2,3], this work outperforms the most advanced bit-serial structures. From the perspective of HW, this work outperforms typical bit-parallel/serial work [1,2] on the FoM (area efficiency × energy efficiency) by 15.7× and 3.38×. The MF-DGCQ allows this work to achieve a 24% performance enhancement and a 4.27× power reduction with a relative error of 3.57% in the typical process corner, which occurs a negligible accuracy loss.

As shown in Fig. 14.6.6, the bit-rotated HCIM macro is fabricated using 28nm CMOS technology. This work achieves a decent accuracy on emerging AI applications: accuracy loss is -1.06% for ResNet-18@ImageNet; -1.75% for ViT@ImageNet; 0.19 for GPT-2@Wikitext-102. For INT8 MAC operations with 32 channel accumulations, the measured access time is 12ns at 0.9V. The peak-energy and area efficiency are 67.8TOPS/W and 1.57TOPS/mm^2. By leveraging a bit-rotated feature-in scheme, this work proposes the most balanced hybrid computing structure, improving the memory density × area efficiency × energy efficiency FoM by more than 28% in comparison to prior SRAM-CIM macros [1-3,5,7,9]. Figure 14.6.7 shows the die micrograph and a chip summary table.

Acknowledgement:
This work was supported in part by the National Science and Technology Major Project under grant 2022ZD0118902, in part by the National Natural Science Foundation of China under grant 92264203 and 62204036, in part by the Key Research and Development Program of Jiangsu Province under grant BE2023020-1, in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under grant KYCX23_0344, and in part by the Fundamental Research Funds for the Central Universities under grant 2242022k60009. The corresponding author is Xin Si (xinsi@seu.edu.cn).

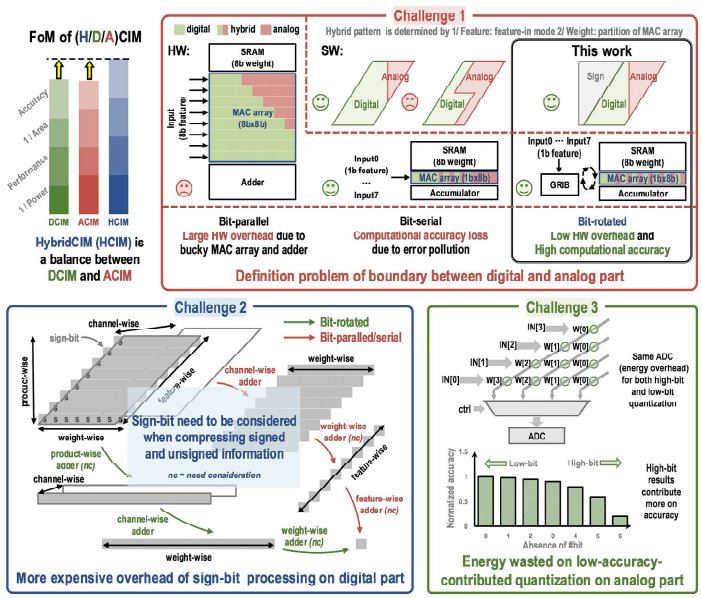


Figure 14.6.1: Hybrid CIM challenges.

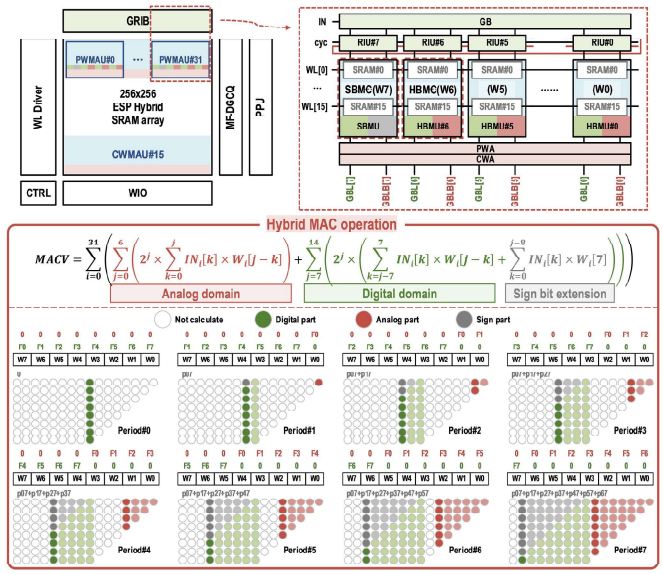


Figure 14.6.2: Overall structure and operations of proposed bit-rotated hybrid CIM macro.

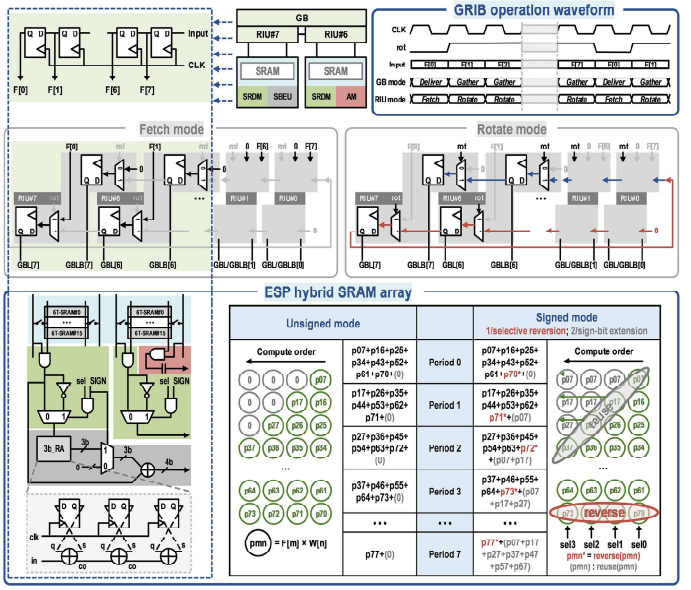


Figure 14.6.3: Detailed schematic and operations of GRIB and ESP hybrid SRAM array.

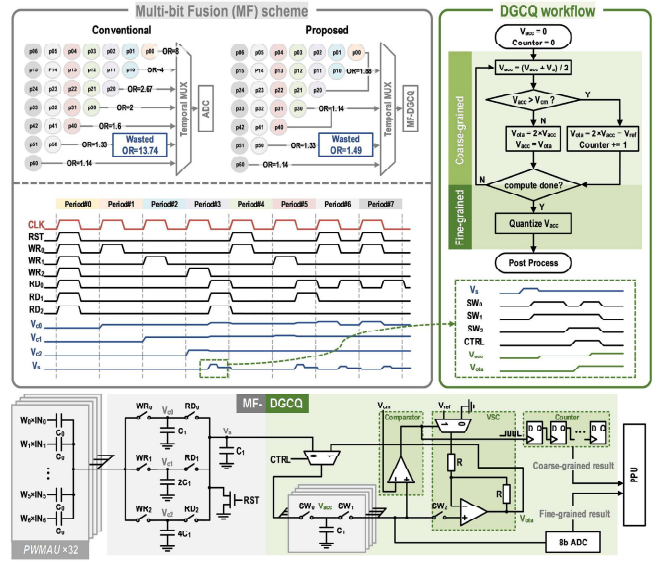


Figure 14.6.4: MF-DGCQ schematic and operations for multi-bit multi-cycle low-power/delay low-bit quantization.

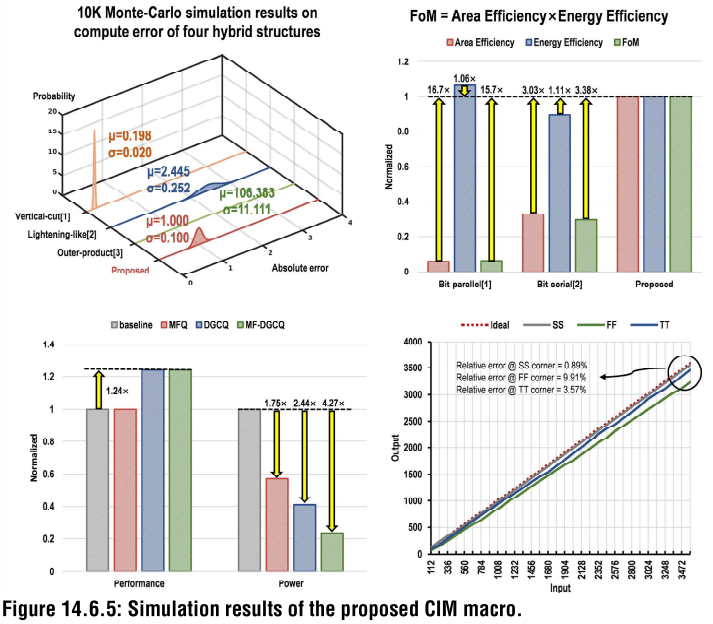


Figure 14.6.5: Simulation results of the proposed CIM macro.

Model	ResNet-18	VIT	GPT-2
Dataset	ImageNet	ImageNet	Wikitext-102
Data precision	INT8	INT8	INT8
Task	Classification	Classification	NLP
Metric	Accuracy	Accuracy	Perplexity
Baseline	69.54%	80.24%	48.39
Proposed	68.48%	78.49%	48.58

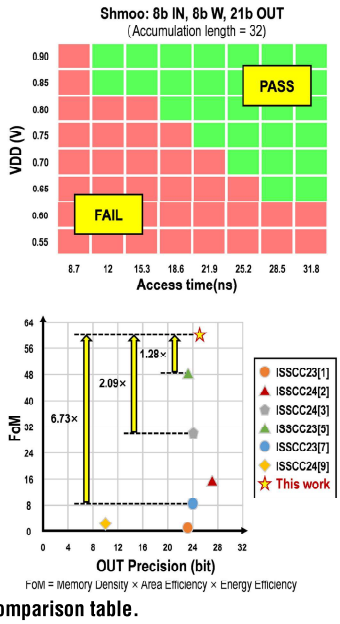
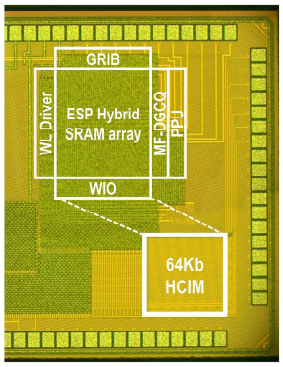


Figure 14.6.6: Measurement results and comparison table.



[1]Measured with worst case under 0.9V. Worst case: 50% feature sparsity, 0% weight sparsity.
[2]Measured under 0.6V.
[3]Software baseline is 69.54%.
[4]Software baseline is 80.24%.
[5]Software baseline is 48.39.

CHIP SUMMARY		
Technology		28nm
Bitcell		6T compact
MAC Domain		Hybrid
Macro Capacity		256×256 (64Kb)
Macro Area (mm^2)		330.19um×328.35um =0.108mm^2
Memory Density (Mb/mm^2)		0.58
Supply Voltage (V)		0.6-0.9
Input precision (bit)		8
Weight precision (bit)		8
Output precision (bit)		21-25
Accumulation length		32-512
Access Time (ns)		12@0.9V
Throughput (TOPS)		[2]0.04-[1]0.17
Area Efficiency (TOPS/mm^2)		[2]0.37-[1]1.57
Energy Efficiency (TOPS/W)		[1]21.04-[2]67.8
Inference Accuracy / Perplexity	[3]ResNet18@ImageNet	-1.06%
	[4]ViT@ImageNet	-1.75%
	[5]GPT-2@Wikitext-102	0.19

Figure 14.6.7: Die micrograph and chip summary table.

References:
[1] P. 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," *ISSCC*, pp. 126-127, 2023.
[2] A. Guo et al., "A 22nm 64kb Lightning-Like Hybrid Computing-in-Memory Macro with a Compressed Adder Tree and Analog-Storage Quantizers for Transformer and CNNs," *ISSCC*, pp. 570-571, 2024.
[3] Y. Yuan et al., "A 28nm 72.12TFLOPS/W Hybrid-Domain Outer-Product Based Floating-Point SRAM Computing-in-Memory Macro with Logarithm Bit-Width Residual ADC," *ISSCC*, pp. 576-577, 2024.
[4] H. Fujiwara et al., "A 5-nm 254-TOPS/W 221-TOPS/mm2 Fully-Digital Computing-in-Memory Macro Supporting Wide-Range Dynamic-Voltage-Frequency Scaling and Simultaneous MAC and Write Operations," *ISSCC*, pp. 186-187, 2022.
[5] 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," *ISSCC*, pp. 128-129, 2023.
[6] H. Mori et al., "A 4nm 6163-TOPS/W/b 4790-TOPS/mm2/b SRAM Based Digital-Computing-in-Memory Macro Supporting Bit-Width Flexibility and Simultaneous MAC and Weight Update," *ISSCC*, pp. 132-133, 2023.
[7] P. Chen et al., "A 22nm Delta-Sigma Computing-In-Memory ($\Delta\sum$CIM) SRAM Macro with Near-Zero-Mean Outputs and LSB-First ADCs Achieving 21.38TOPS/W for 8b-MAC Edge AI Processing," *ISSCC*, pp. 140-141, 2023.
[8] B. Wang et al., "A 28nm Horizontal-Weight-Shift and Vertical-Feature-Shift-Based Separate-WL 6T-SRAM Computation-in-Memory Unit-Macro for Edge Depthwise Neural-Networks," *ISSCC*, pp. 134-135, 2023.
[9] K. Yoshioka et al., "A 818-4094TOPS/W Capacitor-Reconfigured CIM Macro for Unified Acceleration of CNNs and Transformers," *ISSCC*, pp. 574-575, 2024.
[10] S. Hsieh et al., "A 70.85–86.27TOPS/W PVT-Insensitive 8b Word-Wise ACIM with Post-Processing Relaxation," *ISSCC*, pp. 136-137, 2023.

 979-8-3315-4101-9/25/$31.00

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

An Guo*[1], Jingmin Zhang*[1], Xingyu Pu[1], Yi Yang[1], Defa Wu[1], Yuchen Tang[1], Yuhui Shi[1], Yinghai Gao[1], Zhichao Liu[1], Bo Wang[1], Tianzhu Xiong[1], Zhaoyang Zhang[1], Xi Chen[1], Jinwu Chen[1], Feiran Liu[1], Xing Wang[1], Xinning Liu[1], Weiwei Shan[1], Bo Liu[1], Hao Cai[1], Xin Si[1], Jun Yang[1,2]

[1]Southeast University, Nanjing, China
[2]National Center of Technology Innovation for EDA, Nanjing, China

*Equally Credited Authors (ECAs)

Autonomous micro-robots, equipped with AI for urban navigation, are being deployed for various applications, such as package delivery and surveillance. These intelligent machines require efficient path-finding algorithms to navigate complex city environments effectively. Similarly, other domains like VLSI routing, urban path planning, and wavefront propagation also demand optimal navigation solutions, as illustrated in Fig. 14.7.1. While previous work has typically relied on digital accelerators to address navigation challenges [3-5], mimetic-path CIM architectures, which incorporate SRAM within their processing-element (PE) arrays, have demonstrated superior performance, power efficiency, and area utilization (PPA). However, existing mimetic-CIM solutions face several challenges: (1) single-direction searching (SDS) inefficiently utilizes half of the available search time; (2) DFF-based temporary storage suffers from poor PPA, while dynamic-logic based alternatives are prone to leakage issues; and (3) limited processing windows can lead to task failures in large-scale maps, but previous attempts to address these issues, such as map direct transfer and splitting, have their own limitations. This paper presents a 28nm 32×32 mimetic-path NeuroPilot CIM macro with (1) dual-direction searching (DDS) to accelerate the path searching; (2) a dynamic-logic pilot PE with an outspread pre-charge (OPC) circuit, achieving a superior PPA while reducing leakage; and (3) a novel 3-step coarse/fine-path (TsCFP) flow enabling efficient navigation of large maps. The proposed chip achieves a search rate of 3670M nodes/s, 213.6μm² PE area, and a 69.4fJ node energy consumption.

Figure 14.7.2 illustrates the overall structure and operation of the proposed pilot CIM: comprising of a 32×32 pilot PE array, WL and BL drivers, IOs, column and row decoders, an OPC for sequential pre charging, and the main controller. Each pilot PE contains 14 SRAM cells and a PE core connected to 8 adjacent PEs. The proposed dual-direction searching (DDS) shows significant improvement over single-direction searching (SDS), as shown in an ISSCC-like map example in Fig. 14.7.2. DDS initiates paths, starting at both the START and END nodes, meeting at a junction PE; thereby, achieving a 1.13× lower power consumption and a 2.1× faster processing time for the example map. It also achieves a 1.28× power and 2.0× processing improvements for a 32×32 map without block PEs. The NeuroPilot CIM workflow involves map preprocessing, chip controller decision-making for coarse/fine mapping or direct CIM macro processing based on map size. The CIM macro operates in three stages: (1) a memory-write stage, where the map data, including BLOCK (unreachable cells), START and END points, and the traffic-jam level (TJL), are written to the PE SRAM cells; (2) a path-pilot stage, which initiates a path search that runs until convergence, while also storing direction information in PE SRAM cells; and (3) a memory-read stage, where the final path is determined by tracing from the junction PE.

Figure 14.7.3 shows a detailed schematic for the pilot PE, featuring 14b SRAM cells with two types of cells: SRAM_F for storing EN, STA, END, and TJL using a 6T cell and inverter-driver, and SRAM_R for direction data using a 6T cell, local reset, and a inverter-writer. The PE comprises of 8 direction transmitters and a center core which processes the first input signal, IN[2:0], to arrive from adjacent PEs, determines its origin, START or END, and halts further input processing. The three input-signal bits IN[0] – IN[2] represent transmitting signal arrive, signal originates at START, and signal originates at END, respectively. Each transmitter handles its corresponding inputs, transmits the opposite outputs, and stops receiving other inputs. Take N as an example, N transmitter handles inputs from NW, N and NE, transmits S outputs. Redundancy in design, via double dynamic logic and extra input signals, mitigate potential PVT variation errors. The TJL circuit uses a delay tree to simulate path speed. For example, if this node has a traffic jam, then the EMA [2:0] will be 111, which means this node has the slowest speed. When two signals, originating at both START and END are received, the PE generates a junction signal to deactivate the PE and produce row and column signals to mark the junction's position. The dynamic logic design chooses MOS capacitance rather than MOM capacitance for its high density.

Figure 14.7.4 illustrates the proposed macro's outspread pre-charging (OPC) flow. Leveraging the PE array's logic propagation delay the OPC selectively pre-charges PEs, starting with the START and END rows and progressively transmits the pre-charging signal to adjacent (up and down) rows. This approach ends the propagation path upon START and END convergence, where the pre-charge signal is slightly faster than PE array signals. The OPC implementation results in significant improvements: a 1.49× hold-time reduction, leading to a 1.35× lower dynamic-logic capacity requirement and a 1.32× smaller area. The design incorporates feedback row and column signals, connected separately, which are processed by column and row encoders to determine the converging row and column. These signals pinpoint the junction; the proposed macro reads the corresponding SRAM and IO transmit directional data to the PC for further processing.

Figure 14.7.5 illustrates the proposed 3-step coarse/fine-path (TsCFP) flow that addresses the limitations of previous large map-conversion methods. This approach overcomes the vagueness issues of direct-map transfers and local-optima problems of split-map transfers. The process unfolds in three stages: (1) generation of the smallest clear coarse map by iteratively adjusting the resolution and scoring based on a path-ratio comparison between the original and converted maps; (2) creation of a coarse map's shortest path by splitting it into 32×32 segments for piloting, via SDS or DDS, with subsequent maps determined by previous END points and the original map's general direction; and (3) fine-map shortest-path generation by splitting the original map along the shortest coarse path and processing segments sequentially. When applied to an example based on a map of San Francisco the proposed flow achieves a 373.82ns time to solution (TTS), a 512.5pJ energy consumption, and a 1279.8L(unit length) path-length solution.

Figure 14.7.6 presents the measured test-chip results. The test chip is fabricated using a 28nm technology using a 32×32 NeuroPilot CIM macro. Dynamic-logic fading analysis demonstrates that the proposed macro is suitable for tasks up to 360L (84ns), encompassing both real-world scenarios (50-100L) and complex maze-like maps (>100L). The proposed macro occupies a 14.5 × 14.7 = 213.6μm² PE area, consumes of 69.4fJ/node, and achieves a search rate of 3670M nodes/s. In comparison to prior path-searching work [1-5], this work improves the search-rate / energy-per-node / PE-size FoM by 31.49 - 2.148e7×. Figure 14.7.7 shows the die photograph and a design summary table; also shown are short-distance and long-distance search examples.

Acknowledgement:
This work was supported in part by the National Science and Technology Major Project under grant 2022ZD0118902; in part by the National Natural Science Foundation of China under grant no. 6506009793, 92264203 and grant 62204036; in part by the Key Research and Development Program of Jiangsu Province under grant BE2023020-1I in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under grant KYCX23_0344; and in part by the Fundamental Research Funds for the Central Universities under grant 2242022k60009. The corresponding author is Xin Si (xinsi@seu.edu.cn).

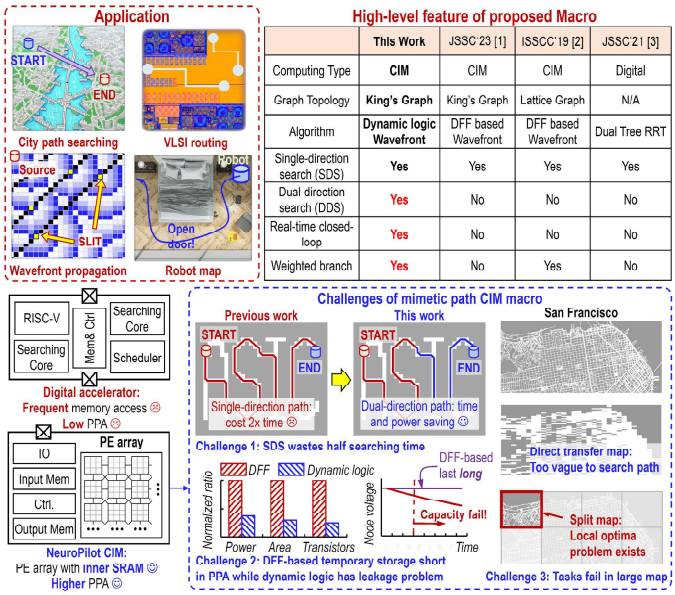


Figure 14.7.1: Motivation and challenges of mimetic path CIMs.

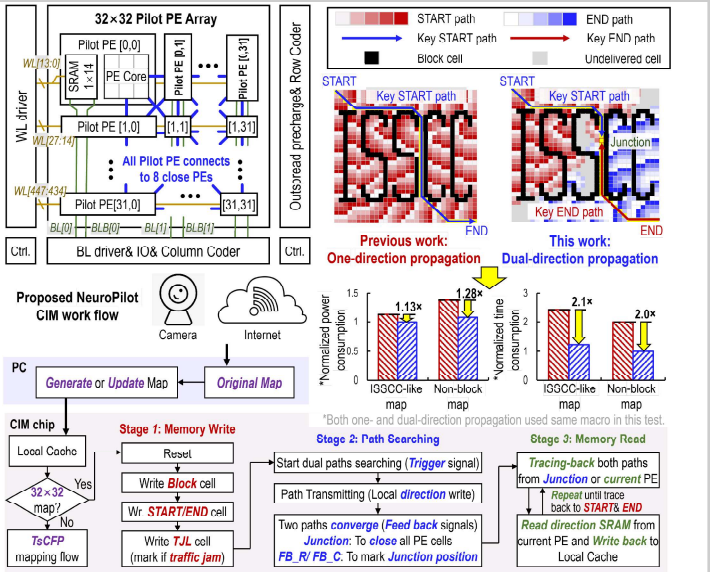


Figure 14.7.2: Proposed NeuroPilot CIM structure, dual-direction search (DDS) propagation and NeuroPilot CIM flow.

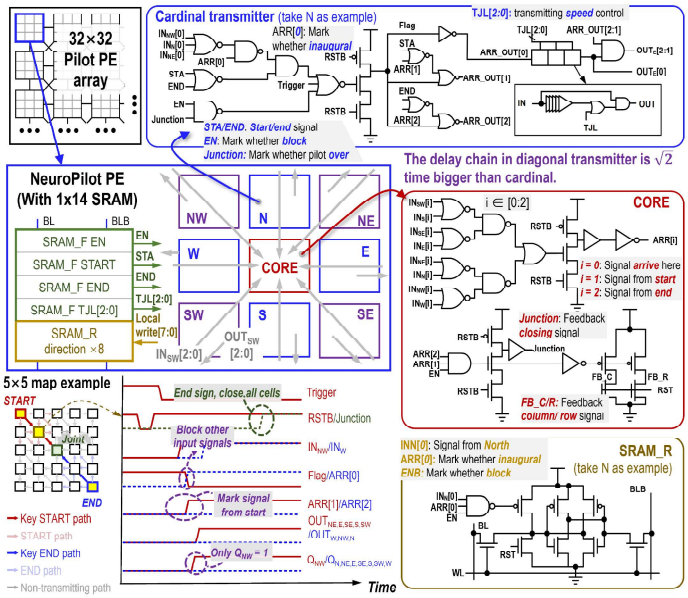


Figure 14.7.3: Detailed pilot PE circuit and operational waveforms.

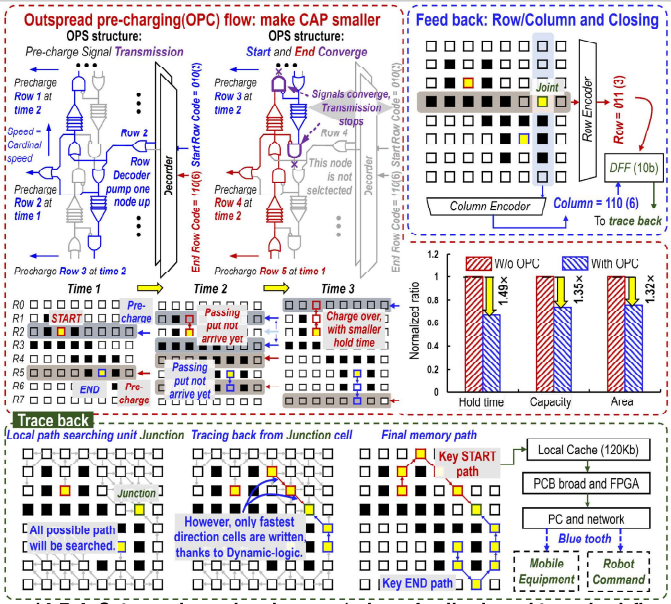


Figure 14.7.4: Outspread pre-charging, row/column feedback, and trace-back flow.

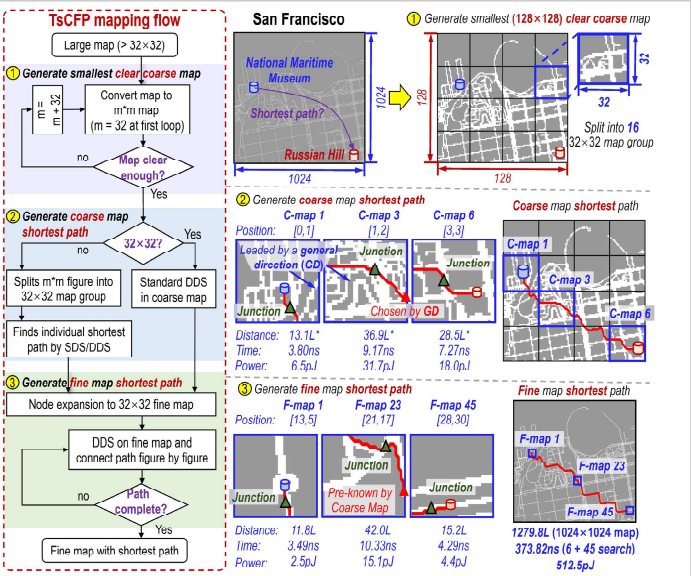
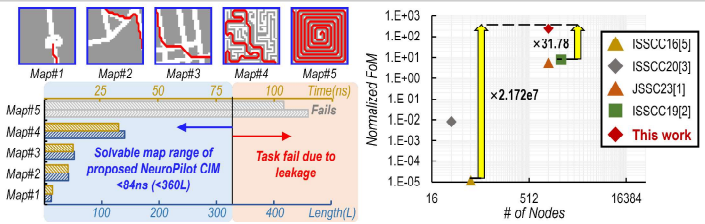


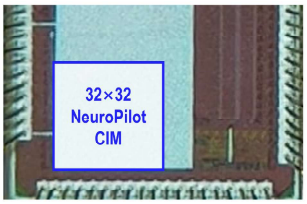
Figure 14.7.5: 3-step coarse/fine-path (TsCFP) mapping flow and an example based on a 1024×1024 map of San Francisco.



SRAM Macro	ISSCC'20 [3]	VLSI'23 [4]	ISSCC'16 [5]	JSSC'23 [1]	ISSCC'19 [2]	This work
Technology	40nm	28nm	65nm	65nm	65nm	28nm
Computing type	Digital	Digital	Digital	CIM	CIM	CIM
Algorithm	Dual Tree RRT	GNN	Iterative Deep. A*	DFF-based Wavefront (SDS)	DFF-based Wavefront (SDS)	Dynamic-logic Wavefront (SDS& DDS)
Supply voltage(V)	0.9	0.9	0.55 - 1.2	1.2/1	1.2	0.9
Core size(mm^2)	1.53×1.53 = 2.34	3.23×1.57 = 5.07	4×4 = 16	0.63×0.62 = 0.39	0.62×0.58 = 0.35	0.48 × 0.47 = 0.228
PE size(μm^2)	N/A	N/A	N/A	19.4×19.8 = 384.12	210.75	14.5 × 14.7 = 213.6
# of Nodes	32	N/A	16×4	32×32	40×40	32×32
Time-to-solution(TTS)	0.125ms	1.79ms	2.13μs	<50ns	N/A	11.4[1]~25.3[2]ns
Node Delay	3.81ns	N/A	33.3ns	0.5/0.75ns	1.79ns	0.45/0.64ns
Normalized Node Delay	3.81ns	N/A	33.3ns	0.5/0.75ns	1.79ns	0.22/0.32ns(Due to DDS)
Energy per task	1.5μJ	330.4μJ	79nJ	776.2pJ	524.8pJ	15.53[1]~71.06[2]pJ
Energy per node	150pJ	N/A	1.23nJ	0.758pJ	0.328pJ	0.0694pJ
Search Rate	262M	N/A	3M	1600M	559M	3670M
FoM[3]	8.17e-3[4]	N/A	1.14e-5[4]	5.50	7.79	247.57

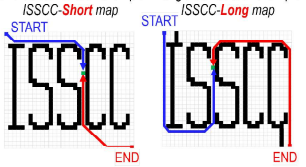
[1]Average test on short distance(<50L) task; [2]Average test on long distance(>100L) task;
[3]FoM=Search Rate × 1/Energy per node × 1/PE size [4]PE size using smallest PE size of all works.

Figure 14.7.6: Measurement results and comparison table.



[1]Average test on short distance(<50L) tasks:
[2]Average test on long distance(>100L) tasks.

[1]Short distance task example. [2]Long distance task example:



CHIP SUMMARY	
Technology	28nm CMOS
Bitcell rule	Logic rule
Cell structure	SRAM_F: 6T + INV
	SRAM_R: 6T + Local Write + Reset
# of Nodes	32 × 32
Macro size	482.51 × 473.38 = 0.228mm^2
PE size	14.5 × 14.7 = 213.6μm^2
Supply voltage(V)	0.9
Time-to-solution(TTS)	11.4[1]~25.3[2]ns
Node Delay	0.45/0.64ns [0.22/0.32ns(Due to DDS)]
Energy per node	0.0694pJ
Search Rate	3670M

Figure 14.7.7: Die photograph and key-metric summary.

References:
[1] C. Yu et al., " A Time-Domain Wavefront Computing Accelerator With a 32 × 32 Reconfigurable PE Array," *IEEE JSSC*, vol. 58, no. 8, pp. 2372-2382, Aug. 2023.
[2] L. Everson et al., " A 40×40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control," *ISSCC*, pp. 50-52, 2019.
[3] C. Chung et al., "A 1.5μJ/Task Path-Planning Processor for 2D/3D Autonomous Navigation of Micro Robots," *ISSCC*, pp. 324-326, 2020.
[4] S. Song et al., "GPPU: A 330.4-μJ/task Neural Path Planning Processor with Hybrid GNN Acceleration for Autonomous 3D Navigation" *IEEE VLSI*, 2023.
[5] Y. Kim et al., " A 0.55V 1.1mW Artificial-Intelligence Processor with PVT Compensation for Micro Robots," *ISSCC*, pp. 258-260, 2016.
[6] L. Everson et al., "A Time-Based Intra-Memory Computing Graph Processor Featuring A* Wavefront Expansion and 2-D Gradient Control," *IEEE JSSC*, vol. 56, no. 7, pp. 2281-2290, July 2021.

979-8-3315-4101-9/25/$31.00