A Fully Bit-Flexible Computation in Memory Macro Using Multi-Functional Computing Bit Cell and Embedded Input Sparsity Sensing

Chun-Yen Yao, Tsung-Yen Wu, Han-Chung Liang, Yu-Kai Chen, and Tsung-Te Liu, *Member, IEEE*

***Abstract*— Computation in memory (CIM) overcomes the von Neumann bottleneck by minimizing the communication overhead between memory and process elements. However, using conventional CIM architectures to realize multiply-accumulate operations (MACs) with flexible input and weight bit precision is extremely challenging. This article presents a fully bit-flexible CIM design with a compact area and high energy efficiency. The proposed CIM macro employs a novel multi-functional computing bit cell design by integrating the MAC and the A/D conversion to maximize efficiency and flexibility. Moreover, an embedded input sparsity sensing and a self-adaptive dynamic range (DR) scaling scheme are proposed to minimize the energy-consuming A/D conversions in CIM. Finally, the proposed CIM macro implementation utilizes an interleaved placement structure to enhance the weight-updating bandwidth and the layout symmetry. The proposed CIM design fabricated in standard 28-nm CMOS technology achieves an area efficiency of 27.7 TOPS/mm^2 and an energy efficiency of 291 TOPS/W, demonstrating a highly energy-area-efficient flexible CIM solution.**

***Index Terms*— Area-efficient, bit scalability, computation in memory (CIM), deep neural network (DNN), energy-efficient, in-memory A/D conversion, sparsity sensing.**

I. Introduction

MODERN machine learning (ML) algorithms, such as deep neural networks (DNNs), require substantial parameters and computations mainly composed of high-dimensional multiply-accumulate operations (MACs). The corresponding hardware implementations have become practical because of the higher computing bandwidth and transistor density with the advancement of semiconductor technology. However, further improvement in energy efficiency using traditional von Neumann architecture is limited, since the computation and storage are separate, and the corresponding data access has dominated the overall energy consumption [1]. This "memory wall" has become the performance bottleneck of efficient implementations for ML algorithms and applications.

Computation in memory (CIM) is proposed to deal with the memory bottleneck by maximizing the utilization rate of the stored data with massively parallel and local processing inside the memory macro. As a result, the CIM architecture can achieve over 10× higher energy efficiency than the state-of-the-art digital accelerators [2]. A typical CIM is commonly accomplished by storing the weight parameters in the memory and then feeding the input activations into the CIM macro to generate the corresponding MAC outputs. This approach is suitable for applications that require only fixed weight and input bit precision [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] but fails to serve the applications demanding the MACs with flexible bit precision. To solve this issue, several CIM works split the weight parameters and input activations into bit groups with different weighting representations for low-bit MACs first. These partial MAC results are then processed by MAC aggregation or near-memory computing (NMC) circuitry to complete full-precision MACs [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. Based on this design principle, the CIM design employing 1-b × 1-b MACs together with MAC aggregation offers a promising solution to realize fully bit-flexible multi-bit MACs, which can be explained by the equation below:

$$y = \mathbf{x} \cdot \mathbf{w} = \sum_{n=0}^{N-1} x_n w_n = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} (-1)^k 2^{p+q} \sum_{n=0}^{N-1} x_n[p] w_n[q] \tag{1}$$

where $\mathbf{x}$ is an N-dimensional input vector whose bit precision in each scalar term x_n is P, $\mathbf{w}$ is an N-dimensional weight vector whose bit precision in each scalar term w_n is Q, and k is an integer term to handle negative conditions for two's complement operations. By (1), it is clear that a general MAC consists of only two classes of components: common terms (summation, $\sum_{n=0}^{N-1} x_n[p] w_n[q]$) and scaling terms ($(-1)^k 2^{p+q}$). As a result, the CIM design utilizing a 1-b × 1-b MAC scheme to compute the common terms can

Manuscript received 18 February 2022; revised 24 July 2022; accepted 16 November 2022. Date of publication 9 January 2023; date of current version 25 April 2023. This article was approved by Associate Editor Kathryn Wilcox. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 110-2218-E-002-034-MBK and Grant MOST 111-2218-E-002-018-MBK; in part by the Intelligent and Sustainable Medical Electronics Research Fund in National Taiwan University; and in part by MediaTek Inc. under Contract MTKC-2022-0125. *(Corresponding author: Tsung-Te Liu.)*

Chun-Yen Yao was with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan. He is now with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: cyyao@eecs.ee.ntu.edu.tw).

Tsung-Yen Wu was with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan. He is now with MediaTek Inc., Taipei 11491, Taiwan (e-mail: r08943051@ntu.edu.tw).

Han-Chung Liang, Yu-Kai Chen, and Tsung-Te Liu are with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: r10943006@ntu.edu.tw; yukai030405@gmail.com; ttliu@ntu.edu.tw).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2022.3224363.

Digital Object Identifier 10.1109/JSSC.2022.3224363

achieve full bit flexibility if all scaling terms are computed via MAC aggregation.

Several recent works [13], [15], [16], [19], [20] have taken advantage of 1-b × 1-b MACs with MAC aggregation to maximize the bit flexibility. Among these designs, analog CIM approaches can potentially achieve a compact CIM macro area without bulky digital adders required in digital CIM counterparts. However, the analog CIM architecture demands a multi-bit A/D conversion process to reconstruct the mixed-signal MAC results back to digital codes. Due to the required A/D conversion and multi-level reference circuitry, its efficiency can be significantly limited. Moreover, the A/D conversion process consumes a significant amount of energy, seriously impacting the CIM energy efficiency. Finally, since the completion of an ML task requires both standard memory access and CIM operations, the performance of standard memory access is also critical. Although the standard write access and the CIM operation use the same number of rows, a single write is not a multi-row access function, causing low weight-updating bandwidth and severe performance degradation of the system latency.

To overcome the design challenges above, this work proposes a fully bit-flexible CIM macro with the following design features.

1) A highly compact CIM computing bit cell (CIMC) can support standard read/write access, 1-b × 1-b MAC, reference voltage generation, and in-memory A/D conversion to maximize the area efficiency by reducing the A/D and reference circuitry overheads.
2) An embedded input sparsity sensing and an automatic on-chip reference voltage generation scheme can realize a self-adaptive dynamic range (DR) scaling based on the real-time input sparsity characteristics to minimize the expensive A/D conversions and maximize the energy efficiency.
3) An interleaved CIMC placement structure can simultaneously accelerate the weight-updating process, maintain the symmetric layout implementation, and support the ping-pong operation for higher weight-updating bandwidth.

The proposed fully bit-flexible CIM macro was implemented and verified in standard 28-nm CMOS technology. The proposed CIM design achieves an area efficiency of 27.7 TOPS/mm^2, representing an 8.15× improvement compared with the previous work. Moreover, the proposed embedded input sparsity sensing and the self-adaptive DR scaling minimize the expensive A/D conversions, realizing 27.4%–30.2% energy reduction and measured peak energy efficiency of 383 TOPS/W. Finally, the proposed interleaved CIMC placement topology can further enable a 32.7% reduction in operating cycles, substantially improving the system latency performance.

This article is organized as follows. Section II introduces the previous bit-flexible CIM works. Section III describes the proposed CIM architecture and the operating principle of in-memory A/D conversion. Section IV introduces the proposed embedded input sparsity sensing and self-adaptive DR scaling techniques. Section V describes the proposed interleaved CIMC placement. The chip implementation and measurement results are shown in Section VI, where comparisons with the state-of-the-art results are also provided. Section VII concludes this article.

II. Related Works

The CIM designs employing 1-b × 1-b MACs with MAC aggregation can be mainly classified into two categories according to its computing scheme: current-based computation [13], [16] and charge-based computation [15]. The current-based CIM design performs one 1-b × 1-b MAC by summing the total discharging currents on the bitlines. This computing scheme features a short CIM delay that benefits from the short bitline discharging time but suffers from high nonlinearity. Okumura et al. [13] proposed a 17T ternary bit cell that consists of two standard 6T SRAM cells and a 5T discharging circuit. The multi-level reference circuitry for the following successive-approximation register (SAR) A/D operation was realized via the binary-weighted reference cells replicating the same discharging path of ternary bit cells. However, only one of the four banks in the MAC operating block can simultaneously access these area-consuming reference cells, significantly deteriorating the memory utilization and the overall area efficiency. Besides, Chiu et al. [16] exploited the small footprint of standard 6T SRAM and directly built the discharging paths via the access transistors. The corresponding digital codes can then be reconstructed via a self-timing tracking and sensing scheme through four pairs of replica bitlines. However, this approach requires additional compensation bit cells to ensure the correct MAC function. The required number of compensation cells equals the maximum number of activated WLs, causing a huge area overhead for CIM designs. In addition, the required linear search A/D process slows down the A/D operation and requires more comparisons than a SAR A/D approach, significantly degrading its energy performance.

On the other hand, the charge-based CIM approach demonstrates better linearity than the current-based designs by performing the computations on capacitors. Besides, it features high integration, since the metal-oxide-metal (MOM) capacitors can be placed right above the transistors. Jia et al. [15] exploited this feature and proposed an 8T1C bit cell array for 1-b × 1-b MACs via charge sharing. The outputs are then converted into digital codes via typical capacitor-switching SAR analog-to-digital converters (ADCs) outside the CIM array. However, this approach can cause a large area penalty due to additional sampling capacitors. Moreover, the sample-and-hold process can result in severe voltage swing reduction [11], [19], seriously deteriorating the sense margin of the comparators and the corresponding CIM performance. In summary, the A/D conversion and the reference circuits have clearly become the performance bottleneck for further efficiency improvement in the CIM designs. Therefore, this work tackles this critical issue by proposing a novel multi-functional computing bit cell architecture that integrates the MAC and A/D conversion together to maximize the CIM efficiency. In addition, the energy-consuming A/D conversions are minimized with the proposed embedded input sparsity sensing and self-adaptive DR scaling to further enhance the CIM energy efficiency.

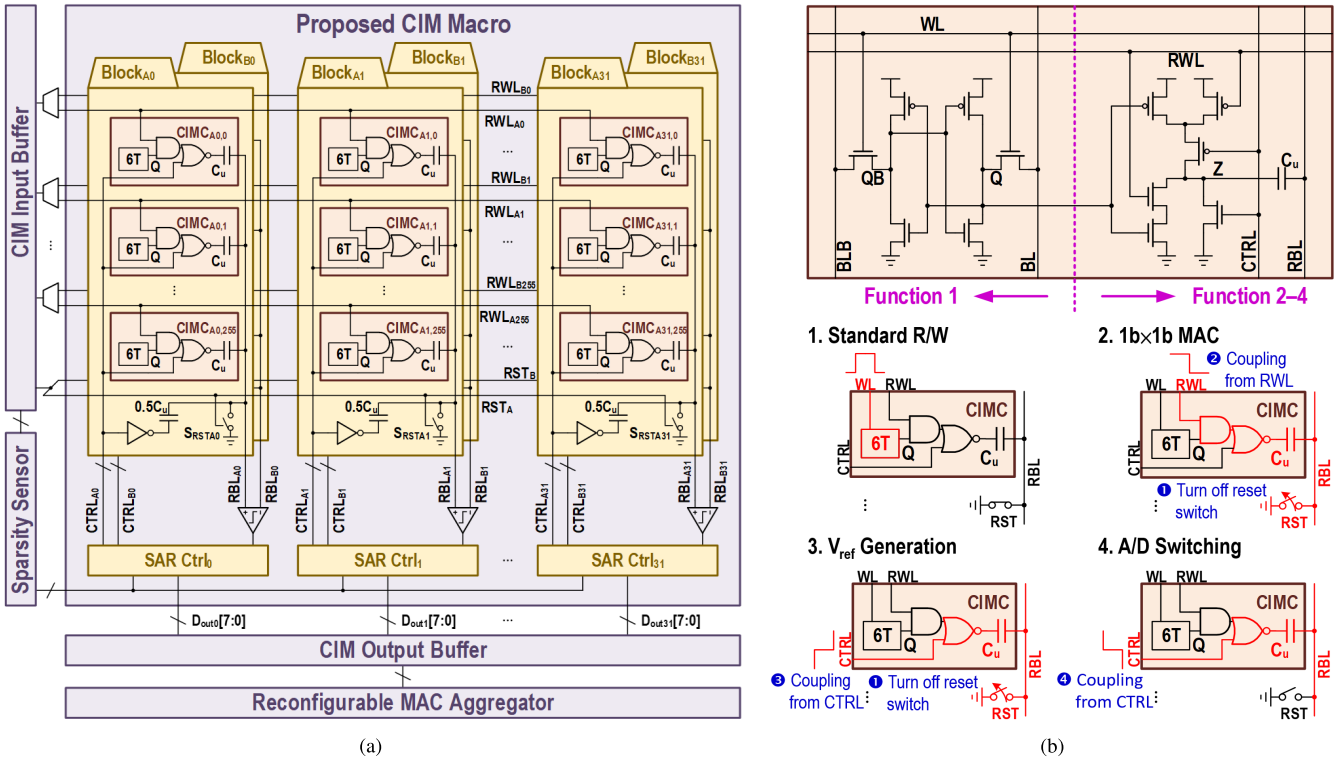


Fig. 1. (a) Overall architecture of the proposed CIM design. (b) Schematic of CIMC and its supporting functions.

III. Proposed CIM Architecture

Fig. 1(a) shows the overall architecture of the proposed bit-flexible CIM architecture with the size of 16-kb CIMCs and the reconfigurable MAC aggregator. The proposed CIM macro is partitioned into 32 CIM block pairs, each of which supports a 256-D 1-b × 1-b MAC. The MAC result can be directly reconstructed back to digital codes through the in-memory A/D conversion process. Moreover, a sparsity sensor is implemented to analyze the real-time CIM input characteristic. This further reduces the CIM energy consumption with the proposed self-adaptive DR scaling, which will be explained in detail in Section IV. After the CIM macro completes its operation, the computed results are sent to the reconfigurable MAC aggregator. The reconfigurable MAC aggregator then performs parallel and sequential shift-add computation to support parallel weights (4/8/16 b) and bit-serial inputs.

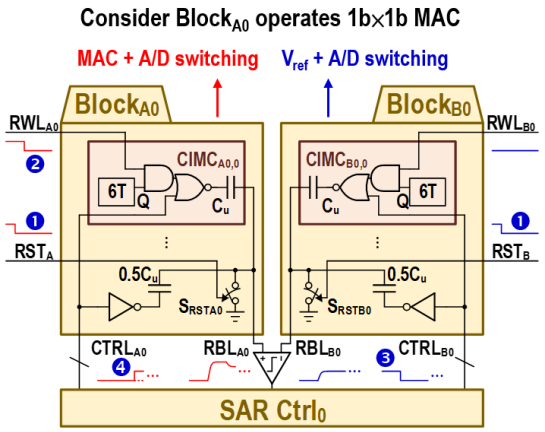


Fig. 2. Operating example of the proposed CIMC 1-b × 1-b MACs in Block_{A0}.

A. Multi-Functional CIMC

The proposed CIMC, as shown in Fig. 1(b), consists of a 6T SRAM for weight storage, a stacked MOM capacitor (C_u) above the transistors, and a 6T AND–OR–INV gate serving as computation logic and capacitor driver. Based on different ways to activate the control signals, including WL, RWL, CTRL, and RST, each CIMC can support the following four functions: 1) standard read/write access; 2) 1-b × 1-b mixed-signal MAC; 3) reference voltage generation; and 4) SAR A/D capacitor switching. As a result, the proposed highly integrated bit cell design can realize the required CIM functions within a compact cell structure. In this way, we avoid additional ADC and reference circuit overhead in conventional CIM architectures and significantly reduce the overall area by 24.1%, as described in Section VI.

An operation of the proposed CIM consists of two steps. Fig. 2 illustrates an operation example of a 256-D 1-b × 1-b MAC computation in Block_{A0}. Fig. 3 shows the corresponding timing diagram. In the first step, Block_{A0} computes the mixed-signal MAC, while Block_{B0} generates the reference voltage V_{ref}. The capacitive coupling mechanism in both blocks can then be expressed as follows:

$$V_{\text{RBL}} = \frac{C_u V_{\text{DD}}}{256C_u + C_{\text{RBL}}} \sum_{i=0}^{255} Z_i \tag{2}$$

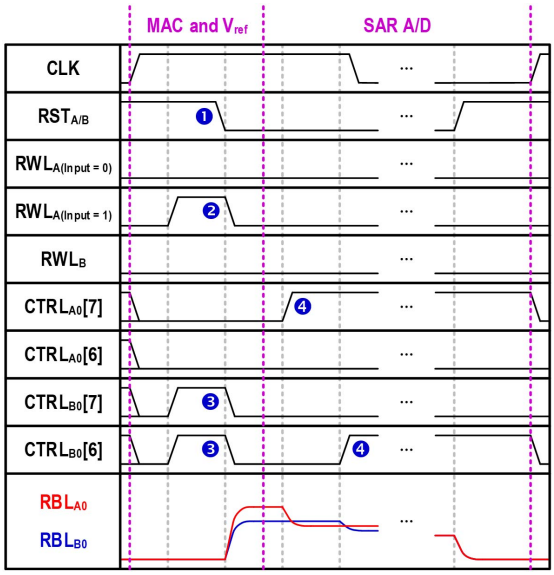


Fig. 3. Timing diagram of the proposed CIMC 1-b × 1-b MACs.

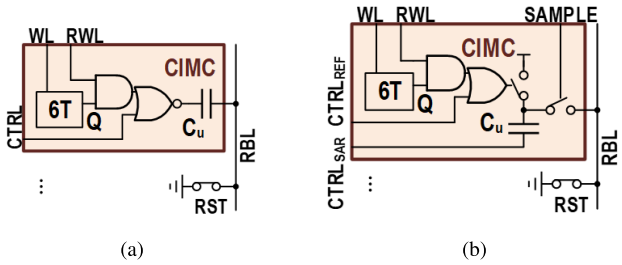


Fig. 4. (a) Proposed bottom-plate sampling CIMC. (b) Alternative top-plate sampling CIMC.

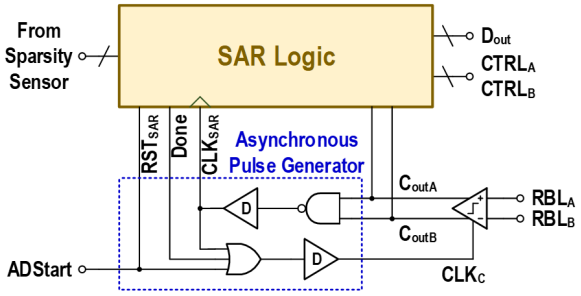


Fig. 5. Schematic of SAR Ctrl block with the comparator.

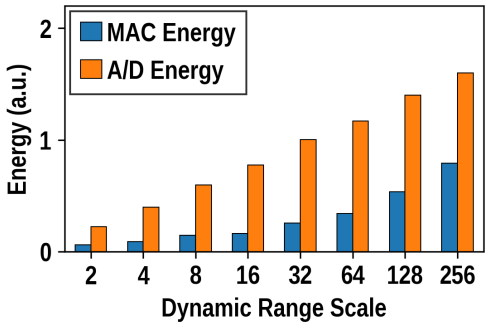


Fig. 6. Simulated energy consumption given different DR scales.

where C_u is the capacitance of C_u, C_{RBL} is the load capacitance on $\mathrm{RBL_{A0}}$ or $\mathrm{RBL_{B0}}$, and Z_i is the cell i whose node Z [annotated in Fig. 1(b)] is pulled to the ground ($Z_i = 1$) or not ($Z_i = 0$) in this step. After the activation pulse is sent from RWL or CTRL, the corresponding MAC result and reference voltage level are capacitively coupled on $\mathrm{RBL_{A0}}$ and $\mathrm{RBL_{B0}}$ in $\mathrm{Block_{A0}}$ and $\mathrm{Block_{B0}}$, respectively. After that, both $\mathrm{Block_{A0}}$ and $\mathrm{Block_{B0}}$ together perform an in-memory SAR A/D conversion, as shown in Fig. 3.

B. In-Memory SAR A/D Conversion

As discussed in Section II, A/D conversion and reference circuits become the performance bottleneck in the CIM designs. Therefore, the proposed CIM design minimizes this overhead by reusing the capacitors in the CIM blocks for A/D conversion and embedded reference voltage generation. The proposed in-memory SAR A/D conversion architecture employs several design techniques to optimize its performance. First, it uses the monotonic switching procedure to minimize the circuit complexity. The resulting optimized CIMC, which realizes MAC, V_{ref} generation, and A/D switching functions, requires only additional six transistors, as shown in Fig. 1(b). Moreover, the asynchronous timing method is employed in SAR A/D conversion to further enhance the throughput performance [26]. Finally, the proposed design uses the bottom-plate sampling topology, as shown in Fig. 4(a). Compared with the top-plate sampling counterpart shown in Fig. 4(b), the proposed design exhibits lower circuit complexity with fewer control signals, leading to a compact CIMC area. In addition, the top-plate sampling approach incurs higher parasitic capacitance, since the node RBL connects to more switches. This can severely degrade the voltage swing and the performance of A/D conversion.

Fig. 5 shows the schematic of the SAR Ctrl block in Fig. 2 with a comparator. The comparator employs a StrongARM topology similar to the design in [27]. The SAR Ctrl block consists of an asynchronous pulse generator for local timing control and a SAR Logic module. After the ADStart signal is triggered at the negative edge, the SAR Ctrl block generates its local timing pulses and performs the corresponding logic operation. At each positive edge of $\mathrm{CLK_{SAR}}$, the registers in the SAR Logic are updated to generate the next CTRL bus configurations. After the conversion process finishes, all of the local timing pulses are disabled by the Done signal, and the final digital output $\mathrm{D_{out}}$ is available at this time.

IV. Embedded Input Sparsity Sensing and Self-Adaptive DR Scaling

To improve the CIM energy efficiency, the proposed design further exploits input characteristics to reduce the CIM energy consumption resulting from the A/D conversion process. Fig. 6 compares the simulated MAC and A/D energy as a function of different DR scales for a uniformly distributed input at 0.8 V. This result clearly demonstrates that the A/D conversion dominates the total energy consumption in a CIM operation. However, the CIM computation results seldom reach the whole

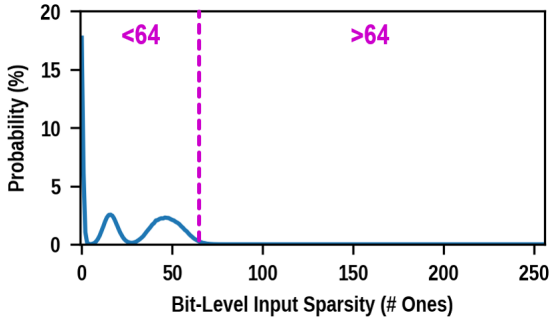


Fig. 7. Modeled input sparsity of CIFAR-10 task with ResNet-18.

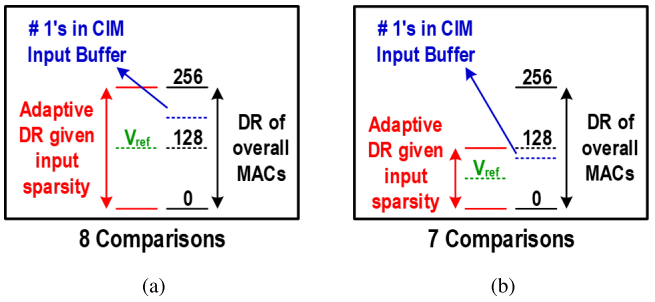


Fig. 9. Concept of self-adaptive DR scaling given (a) dense and (b) sparse input.

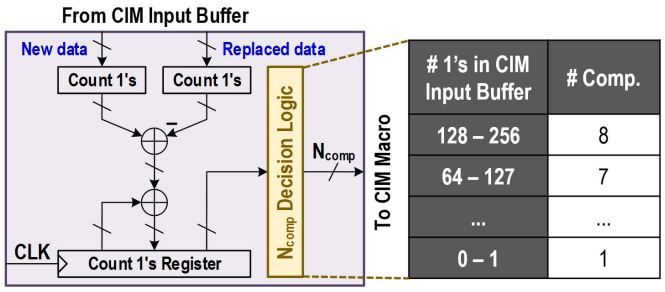


Fig. 8. Proposed sparsity sensor.

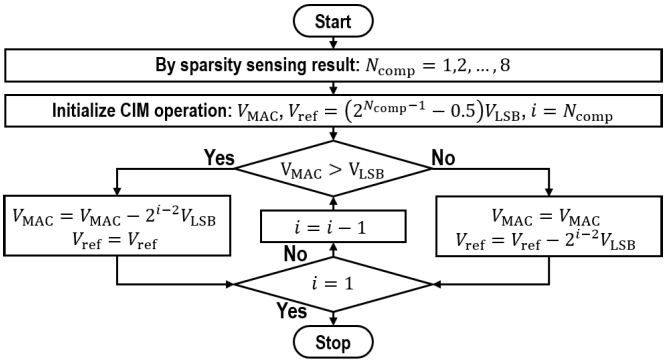


Fig. 10. Proposed automatic DR scaling algorithm.

DR of MACs, especially when the input data characteristic is sparse. Fig. 7 illustrates the modeled probability distribution of the bit-level input sparsity for a typical CIFAR-10 classification task with ResNet-18. The value of input sparsity mainly resides in the range where the bit-level input sparsity is lower than 64. This distribution suggests that the corresponding DR could be lower for input with high sparsity, and the number of voltage comparisons N_{comp} can be reduced accordingly.

The information of the input sparsity can actually be analyzed in advance during the run time, since these data must be filled into the CIM input buffer shown in Fig. 1 before entering the CIM macro. With the real-time sparsity information, N_{comp} can, thus, be minimized accordingly during run time to reduce the overall energy consumption. Therefore, a self-adaptive DR scaling is proposed to exploit this characteristic by using an embedded bit-serial input sparsity sensor together with an automatic voltage generation scheme.

The proposed input sparsity sensor shown in Fig. 8 can estimate the real-time input sparsity by detecting the change of the number of ones whenever new data are fed into the CIM input buffer. Then, different reference voltage generation levels are automatically configured according to the estimated input sparsity characteristic. Fig. 9 illustrates the concept of the self-adaptive DR scaling scheme given different input sparsity characteristics. The reference voltage level is generated to half of the adaptive DR to ensure functionality. N_{comp} is then determined to realize a sparsity-aware, energy-efficient A/D conversion. Fig. 9(b) illustrates that the optimal DR and N_{comp} can be reduced accordingly for a sparse input to minimize the A/D conversion energy.

The self-adaptive DR scaling algorithm is shown in Fig. 10. Given the information of N_{comp} estimated by the sparsity sensor, the corresponding reference voltage V_{ref} can be generated by setting up the number of switched capacitors according to (2) and activating the CTRL bus shown in Fig. 2 during the step of reference voltage generation. In the following step, the in-memory SAR A/D conversion is completed through the asynchronous monotonic switching process described in Section III-B.

V. Interleaved CIMC Placement

Both standard memory access and CIM operations can impact the performance of CIM executing an ML task. Conventional standard write access and CIM computation use the same number of rows. However, a single write is not a multi-row access function, causing a low weight-updating bandwidth and a substantial increase in the system latency. Therefore, this work proposes an interleaved CIMC placement structure shown in Fig. 11 to accelerate the weight-updating process. The proposed structure shapes the 256 CIMCs into two columns with interleaved RWLs for CIM MAC access. In this way, the CIMCs connected to one WL are doubled compared with the conventional designs without interleaving the RWLs. As a result, the corresponding write speed can be twice faster than the conventional non-interleaved design. Moreover, the RBL can be sandwiched between the symmetric capacitor array to maximize the axial symmetry with the proposed placement structure. In this way, the impact of random mismatch and the non-ideal coupling effect from other CIM blocks can be minimized. Fig. 12 shows the CIM performance improvements using the proposed interleaved placement structure with three types of DNN implementations

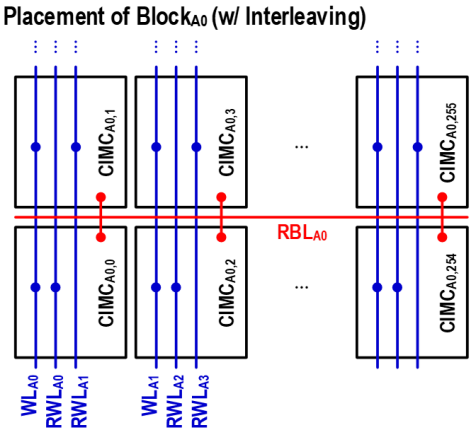


Fig. 11. Interleaved CIMC placement, where the dots denote the wire access to the CIMCs.

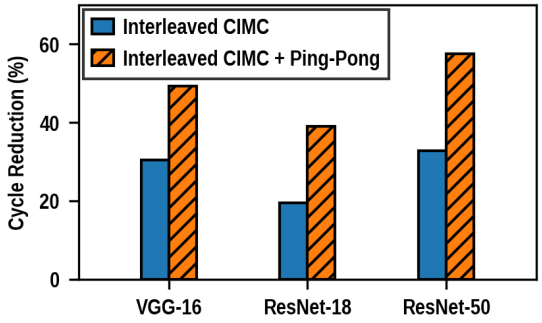


Fig. 12. Operating cycle reduction using the proposed interleaved CIMC placement structure.

for a CIFAR-10 task. The proposed approach can significantly reduce the overall operating cycles by 30.4%, 19.5%, and 32.7% in VGG-16, ResNet-18, and ResNet-50 topology, respectively.

In addition, the proposed CIMC and the interleaved placement structure can readily support the efficient ping-pong operation [22], which realizes CIM and weight-updating operations at the same time. As shown in Fig. 1(b), the weight storage in the proposed CIMC can be decoupled from the AND–OR–INV gate when the RWL is not activated. In this way, the CIM blocks that do not perform MAC operations can update their stored weights simultaneously when performing the reference voltage generation or the SAR A/D switching operation, as illustrated in Fig. 13. Overall, the operating cycles can be further reduced by 49.3%, 39.0%, and 57.6% in VGG-16, ResNet-18, and ResNet-50 topology, respectively, as shown in Fig. 12.

VI. Chip Implementation and Measurement Results

The proposed CIM architecture was designed and implemented with standard 28-nm CMOS technology. Fig. 14 shows the die photograph of the CIM test chip. Table I summarizes the measured CIM performance results with different inputs and weight precision. By integrating several critical CIM functions, the bit cell area of the proposed 12T CIMC

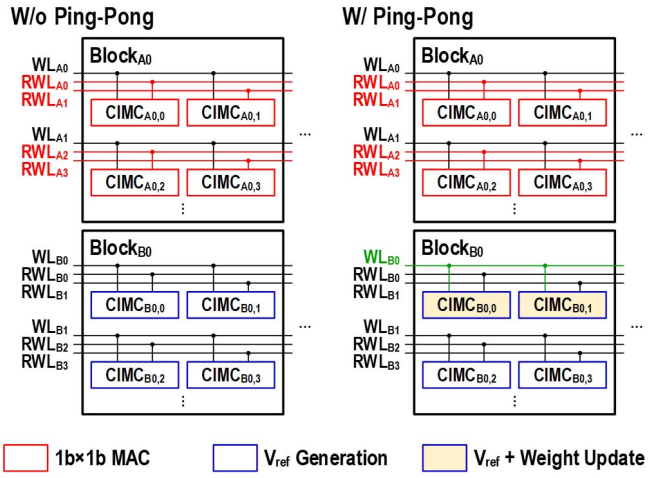


Fig. 13. Ping-pong operation using the proposed CIM macro.

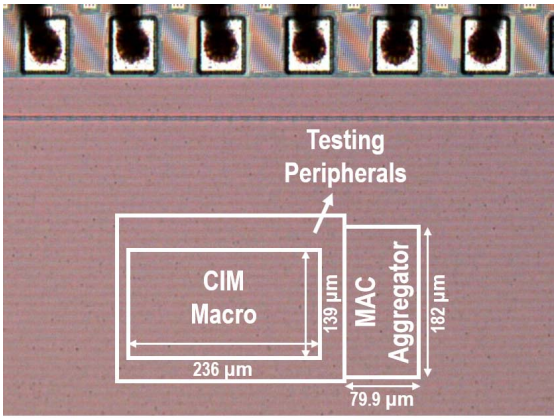


Fig. 14. Die photograph of the proposed CIM design.

TABLE I
Chip Performance Summary

CIM Size	16 kb			
Bit Cell Area	1.215 µm^2			
CIM Macro Area	32900 µm^2			
MAC Aggregator Area	14500 µm^2			
Core Area (Macro+Aggregator)	47400 µm^2			
Supply	0.7 V / 0.8 V			
Clock Rate	50 MHz / 80 MHz			
Input / Weight Bits	1 / 1	4 / 4	8 / 8	16 / 16
Latency (ns)	20 / 12.5	100 / 62.5	180 / 112.5	340 / 212.5
Throughput (GOPS)	819 / 1310	51.2 / 81.9	12.8 / 20.5	3.20 / 5.12
Core Energy Efficiency (TOPS/W)	383 / 291	23.9 / 18.2	5.98 / 4.55	1.50 / 1.14
Core Area Efficiency (TOPS/mm^2)	17.3 / 27.7	1.08 / 1.73	0.270 / 0.433	0.0676 / 0.108

demonstrates a compact footprint of 1.215 μm^2 with an area overhead of 66.7% compared with the standard 6T SRAM cell. As a result, the proposed design achieves a high area efficiency

TABLE II
PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART CIM DESIGNS

	VLSI'19 [13]	JSSC'20 [15]	JSSC'20 [16]	JSSC'21 [19]	**This Work**
Tech. (nm)	12	65	55	65	**28**
CIM Size (kb)	4740	576	4	64	**16**
Bit Cell Area (μm^2)	NA	1.8	NA	2.6	**1.215**
Bit Cell CIM Functions	MAC	MAC	MAC + Ref.	MAC	**MAC + Ref. + A/D**
Interleaved Placement	F	F	F	F	**T**
Core Area (mm^2)	9[a]	8.56	5.94[a]	0.175	**0.0473**
Supply (V)	0.72	1.2/0.85	0.9	1.2	**0.8/0.7**
Input Bits	1/4	1–8	1/2/7/8	1–8	**1–16**
Weight Bits	1/4	1–8	1/2/8	1/2/3/4/5/8	**1/4/8/16**
Cycle Time (ns)	10	10 (1.2 V) 25 (0.85 V)	3.5	14.3	**12.5 (0.8 V)** **20 (0.7 V)**
Throughput[b] (TOPS)	0.172	2.185 (1.2 V) 0.874 (0.85 V)	0.3291	0.5734	**1.31 (0.8 V)** **0.819 (0.7 V)**
Energy Efficiency[b] (TOPS/W)	8.8	192 (1.2 V) 400 (0.85 V)	40.2	49.4	**291 (0.8 V)** **383 (0.7 V)**
Area Efficiency[b] (TOPS/mm^2)	0.0191	0.6 (1.2 V) 0.24 (0.85 V)	0.05538	3.4	**27.7 (0.8 V)** **17.3 (0.7 V)**

[a]Only die area is reported.
[b]Normalized to to 1b×1b MACs (1 MAC is 2 OPs).

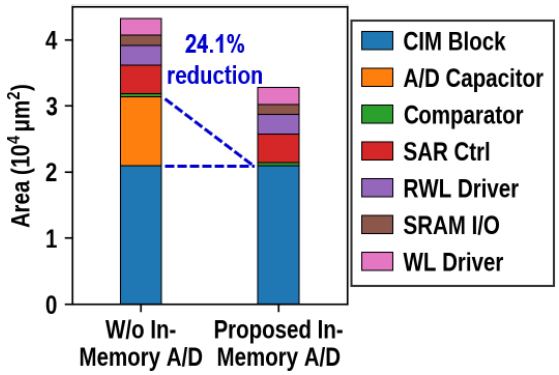


Fig. 15. Area breakdown between the two different CIM macro designs.

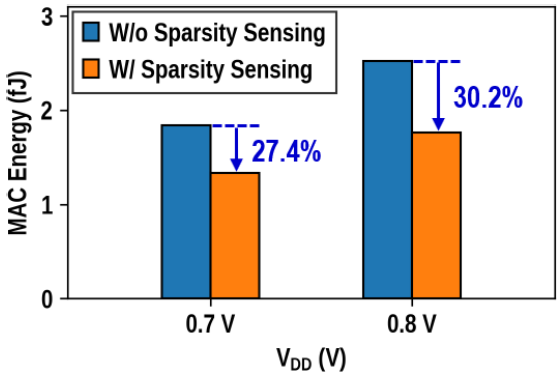


Fig. 16. Measured energy improvement with the proposed embedded input sparsity sensing and self-adaptive DR scaling scheme.

of 27.7 TOPS/mm^2. Moreover, with the proposed embedded input sparsity sensing and self-adaptive DR scaling scheme, our design achieves a peak energy efficiency of 383 TOPS/W. Given the CIFAR-10 dataset, the proposed design can achieve 91% classification accuracy with ResNet-18.

Fig. 15 compares the area breakdown of the proposed CIM macro with the baseline design. The proposed CIM architecture with in-memory A/D conversion significantly decreases the macro area from 43 200 to 32 800 μm^2, representing a 24.1% area reduction. To evaluate the energy improvement with the proposed embedded input sparsity sensing and self-adaptive DR scaling scheme, an additional test mode that can disable the sparsity sensor was implemented in the CIM test chip. Fig. 16 compares the measured MAC energy of the proposed design before and after the embedded sparsity sensor is disabled. For a CIFAR-10 task with ResNet-18 topology, the proposed design with the input sparsity sensing and self-adaptive DR scaling realizes 27.4% and 30.2% energy reduction at 0.7 and 0.8 V, respectively, effectively minimizing the A/D conversion energy to enhance the overall CIM efficiency.

Table II compares the proposed CIM architecture with the state-of-the-art designs that support flexible input and weight bit precision. The proposed multi-functional CIMC implemented in an advanced CMOS technology process enables a highly integrated and compact CIM design, demonstrating 8.15× higher area efficiency and 5.89× higher energy efficiency than the previous work with the highest area efficiency [19]. Moreover, with the proposed embedded input sparsity sensing and self-adaptive DR scaling scheme, the proposed CIM design achieves similar energy efficiency while realizing 72.1× higher area efficiency than the most energy-efficient CIM [15]. Finally, the proposed CIM macro utilizes an interleaved placement structure to maintain symmetric layout implementation and accelerate the weight-updating process, substantially reducing the overall

latency and benefiting large-scale CIM-based computation systems.

VII. Conclusion

This article presents an energy-area-efficient CIM design that can support emerging ML applications with flexible bit precision. The highly compact multi-functional CIMC design with in-memory SAR A/D conversion can maximize the CIM efficiency and flexibility. In addition, the proposed embedded input sparsity sensing and self-adaptive DR scaling scheme minimize the expensive A/D conversions effectively. Finally, the interleaved placement structure is proposed to improve the weight-updating bandwidth and maintain the layout symmetry simultaneously. The measurement results show that the proposed CIM design achieves the high energy and area efficiencies of 291 TOPS/W and 27.7 TOPS/mm^2, respectively, representing a highly efficient bit-flexible CIM solution.

Acknowledgment

The authors would like to thank Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu, Taiwan, and the Taiwan Semiconductor Research Institute (TSRI), Hsinchu, for providing chip fabrication and technical support. They would also like to thank Bing-Chen Wu for providing suggestions to this manuscript.

References

[1] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 10–14.

[2] N. Verma et al., "In-memory computing: Advances and prospects," *IEEE Solid-State Circuits Mag.*, vol. 11, no. 3, pp. 43–55, Summer 2019.

[3] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6T SRAM array," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, Apr. 2017.

[4] S. K. Gonugondla, M. Kang, and N. R. Shanbhag, "A variation-tolerant in-memory machine learning classifier via on-chip training," *IEEE J. Solid-State Circuits*, vol. 53, no. 11, pp. 3163–3173, Nov. 2018.

[5] A. Biswas and A. P. Chandrakasan, "CONV-SRAM: An energy-efficient SRAM with in-memory dot-product computation for low-power convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 217–230, Jan. 2019.

[6] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-tile 2.4-Mb in-memory-computing CNN accelerator employing charge-domain compute," *IEEE J. Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.

[7] H. Kim, Q. Chen, and B. Kim, "A 16K SRAM-based mixed-signal in-memory computing macro featuring voltage-mode accumulator and row-by-row ADC," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2019, pp. 35–36.

[8] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok, "C3SRAM: An in-memory-computing SRAM macro based on robust capacitive coupling computing mechanism," *IEEE J. Solid-State Circuits*, vol. 55, no. 7, pp. 1888–1897, Jul. 2020.

[9] Q. Dong et al., "A 351TOPS/W and 372.4GOPS compute-in-memory SRAM macro in 7 nm FinFET CMOS for machine-learning applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 242–244.

[10] C. Yu, T. Yoo, T. T.-H. Kim, K. C. T. Chuan, and B. Kim, "A 16K current-based 8T SRAM compute-in-memory macro with decoupled read/write and 1–5 bit column ADC," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Mar. 2020, pp. 1–4.

[11] Y.-T. Hsu, C.-Y. Yao, T.-Y. Wu, T.-D. Chiueh, and T.-T. Liu, "A high-throughput energy–area-efficient computing-in-memory SRAM using unified charge-processing network," *IEEE Solid-State Circuits Lett.*, vol. 4, pp. 146–149, 2021.

[12] S. Xie, C. Ni, A. Sayal, P. Jain, F. Hamzaoglu, and J. P. Kulkarni, "eDRAM-CIM: Compute-in-memory design with reconfigurable embedded-dynamic-memory array realizing adaptive data converters and charge-domain computing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 248–250.

[13] S. Okumura, M. Yabuuchi, K. Hijioka, and K. Nose, "A ternary based bit scalable, 8.80 TOPS/W CNN accelerator with many-core processing-in-memory architecture with 896K synapses/mm^2," in *Proc. Symp. VLSI Technol.*, Jun. 2019, pp. 248–249.

[14] X. Si et al., "A twin-8T SRAM computation-in-memory unit-macro for multibit CNN-based AI edge processors," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 189–202, Jan. 2020.

[15] H. Jia, H. Valavi, Y. Tang, J. Zhang, and N. Verma, "A programmable heterogeneous microprocessor based on bit-scalable in-memory computing," *IEEE J. Solid-State Circuits*, vol. 55, no. 9, pp. 2609–2621, Jan. 2020.

[16] Y.-C. Chiu et al., "A 4-Kb 1-to-8-bit configurable 6T SRAM-based computation-in-memory unit-macro for CNN-based AI edge processors," *IEEE J. Solid-State Circuits*, vol. 55, no. 10, pp. 2790–2801, Oct. 2020.

[17] J.-W. Su et al., "A 28 nm 64 Kb inference-training two-way transpose multibit 6T SRAM compute-in-memory macro for AI edge chips," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Sep. 2020, pp. 240–242.

[18] C.-X. Xue et al., "A 22 nm 2 Mb ReRAM compute-in-memory macro with 121–28TOPS/W for multibit MAC computing for tiny AI edge devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 244–246.

[19] Z. Chen et al., "CAP-RAM: A charge-domain in-memory computing 6T-SRAM for accurate and precision-programmable CNN inference," *IEEE J. Solid-State Circuits*, vol. 56, no. 6, pp. 1924–1935, Jun. 2021.

[20] H. Kim, T. Yoo, T. T.-H. Kim, and B. Kim, "Colonnade: A reconfigurable SRAM-based digital bit-serial compute-in-memory macro for processing neural networks," *IEEE J. Solid-State Circuits*, vol. 56, no. 7, pp. 2221–2233, Jul. 2021.

[21] X. Si et al., "A local computing cell and 6T SRAM-based computing-in-memory macro with 8-b MAC operation for edge AI chips," *IEEE J. Solid-State Circuits*, vol. 56, no. 9, pp. 2817–2831, Sep. 2021.

[22] J. Yue et al., "A 2.75-to-75.9TOPS/W computing-in-memory NN processor supporting set-associate block-wise zero skipping and ping-pong CIM with simultaneous computation and weight updating," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 238–240.

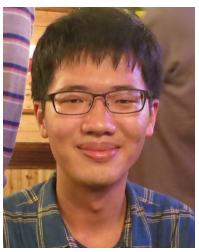
[23] C.-X. Xue et al., "A 22 nm 4 Mb 8b-precision ReRAM computing-in-memory macro with 11.91 to 195.7TOPS/W for tiny AI edge devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 245–247.

[24] J.-W. Su et al., "A 28 nm 384 kb 6T-SRAM computation-in-memory macro with 8b precision for AI edge chips," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 250–252.

[25] Y.-D. Chih et al., "An 89TOPS/W and 16.3TOPS/mm^2 all-digital SRAM-based full-precision compute-in memory macro in 22 nm for machine-learning edge applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 252–254.

[26] S.-W. M. Chen and R. W. Brodersen, "A 6-bit 600-MS/s 5.3-mW asynchronous ADC in 0.13-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2669–2680, Dec. 2006.

[27] Q. Fan, Y. Hong, and J. Chen, "A time-interleaved SAR ADC with bypass-based opportunistic adaptive calibration," *IEEE J. Solid-State Circuits*, vol. 55, no. 8, pp. 2082–2093, Aug. 2020.



Chun-Yen Yao received the B.S. degree in electrical engineering with a minor in mechanical engineering and the M.S. degree in electronics engineering from National Taiwan University, Taipei, Taiwan, in 2019 and 2022, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with the University of California at Berkeley, Berkeley, CA, USA.

His current research interests include low-noise current sensing and adaptive sensing for biomedical applications.

Mr. Yao received the National Taiwan University Outstanding Youth Award in 2021.

Tsung-Yen Wu received the B.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2019, and the M.S. degree in electronics engineering from National Taiwan University, Taipei, Taiwan, in 2022.

He is currently with MediaTek Inc., Taipei. His current research interests include computing in memory for energy-efficient machine learning applications.

Han-Chung Liang received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2021, where he is currently pursuing the M.S. degree with the Graduate Institute of Electronics Engineering.

His current research interests include computation in memory circuit design and energy-efficient circuit design.

Yu-Kai Chen received the B.S. degree in electrical engineering from National Chung Hsing University, Taichung, Taiwan, in 2021. He is currently pursuing the M.S. degree with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan.

His research interests include computation in memory and mixed-signal circuit designs.

Tsung-Te Liu (Member, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in electronics engineering from National Taiwan University, Taipei, Taiwan, in 2002 and 2004, respectively, and the Ph.D. degree in electrical engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2012.

From 2004 to 2005, he was with MediaTek Inc., Hsinchu, Taiwan, where he was involved in circuit and system design for wireless communications. From 2005 to 2012, he was a member of the Berkeley Wireless Research Center (BWRC), University of California at Berkeley. From 2012 to 2014, he was with Interuniversity Microelectronics Centre (IMEC), Leuven, Belgium, where he conducted research on circuit development for advanced CMOS technology. In 2014, he joined the faculty of National Taiwan University, where he is currently an Associate Professor with the Graduate Institute of Electronics Engineering and the Department of Electrical Engineering.

Dr. Liu was a recipient of several design and teaching awards. His research interests involve energy-efficient circuit and system designs.