A 28-nm 36 Kb SRAM CIM Engine With 0.173 μm^2 4T1T Cell and Self-Load-0 Weight Update for AI Inference and Training Applications

Chenyang Zhao, Jinbei Fang, Xiaoli Huang, Deyang Chen, Zhiwang Guo, Jingwen Jiang, Jiawei Wang, Jianguo Yang, *Member, IEEE*, Jun Han, *Member, IEEE*, Peng Zhou, *Senior Member, IEEE*, Xiaoyong Xue, *Member, IEEE*, and Xiaoyang Zeng, *Senior Member, IEEE*

***Abstract*— Computing-in-memory (CIM) promises high energy efficiency (EE) and performance in accelerating the feed-forward (FF) and back-propagation (BP) processes of deep neural networks (DNNs) with less data movement and high parallelism. However, challenges still lie in large memory cells, network mapping, and IR-drop variation to realize efficient CIM implementation. In this work, a 28-nm 36 Kb static random-access memory (SRAM) CIM engine with nondestructive-read (NDR) cell and weight update energy saving is used for multiply-accumulate (MAC) acceleration in artificial intelligence (AI) inference and train applications. A 4T1T SRAM bit-cell is proposed with NDR and records the smallest cell size of 0.173 μm^2. The power-on self-load-0 feature of the 4T1T cell saves the weight update energy and latency for writing 0. The shared-path dual-mode read (SPDMR) brings fewer circuit overheads to support both FF and BP paths. The bit-interleaving weight mapping (BIWM) speeds up the BP path without slowing FF. IR-drop-aware adaptive clampers (IRDAA-Cs) with hierarchical read word-lines (RWLs) and read bit-lines (RBLs) apply possibly accurate voltages on near/far cells. The engine achieves an EE of 263.1/412.1 TOPS/W, as well as an area efficiency (AE) of 2.5/4.9 TOPS/mm^2 for FF/BP process @1-bit weight/activation with 74.4%–78.3% reduction in weight update energy.**

***Index Terms*— 4T1T static random-access memory (SRAM) cell, computing-in-memory (CIM), on-chip training, IR-drop aware architecture, shared-path read, weight update.**

Manuscript received 4 September 2023; revised 23 December 2023 and 26 April 2024; accepted 4 May 2024. Date of publication 21 May 2024; date of current version 26 September 2024. This article was approved by Associate Editor Meng-Fan Chang. This work was supported in part by STI 2030—Major Projects under Grant 2022ZD0209200; in part by the National Natural Science Foundation of China under Grant 62274038, Grant 61934002, and Grant 62222119; in part by the Science and Technology Commission of Shanghai Municipality under Grant 21TS1401200; and in part by the ZTE Industry-University-Institute Cooperation Funds under Grant 1A20230201004. *(Corresponding author: Xiaoyong Xue.)*

Chenyang Zhao, Jinbei Fang, Xiaoli Huang, Deyang Chen, Zhiwang Guo, Jingwen Jiang, Jiawei Wang, Jun Han, and Xiaoyang Zeng are with the State Key Laboratory of Integrated Chips and Systems, School of Microelectronics, Fudan University, Shanghai 200433, China.

Jianguo Yang is with the Laboratory of Microelectronics Devices and Integrated Technology, Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, China.

Peng Zhou and Xiaoyong Xue are with the State Key Laboratory of Integrated Chips and Systems, School of Microelectronics, Fudan University, Shanghai 200433, China, and also with the Shaoxin Laboratory, Shaoxing 312000, China (e-mail: xuexiaoyong@fudan.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2024.3399615.

Digital Object Identifier 10.1109/JSSC.2024.3399615

I. Introduction

THE rapid development of artificial intelligence (AI) has created a surging demand for computational resources, such as central processing units (CPUs), graphics processing units (GPUs), field-programmable gate arrays (FPGAs), and so on. Besides, the substantial parameters associated with deep learning neural networks (DNNs) have led to huge data movement between the memory and the processor in the von Neumann architecture, severely impacting the performance and energy efficiency (EE) of AI systems [3]. Computation-in-memory (CIM) technology configures memory as computational storage units to realize multiply-accumulate (MAC) operations with less data movement, improving both performance and EE [6], [7]. Currently, SRAM-based CIM has gained popularity thanks to the mature memory technology and its compatibility with advanced processes [9]. On-chip training is also becoming necessary to enhance the robustness of trained networks against changing environments [11]. However, there are still several challenges to be dealt with for SRAM CIM.

The huge costs from weight storage and updating have brought significant pressure on CIM hardware design. The volume of parameters of DNNs surges with the algorithm complexity, as shown in Fig. 1(a) [13]. However, the storage density of conventional SRAM is low with at least six transistors (6T) per cell. For the nondestructive read (NDR), the 8T SRAM cell with a decoupled read path brings an additional >30% area compared to the 6T counterpart [14]. Besides, both 6T and 8T SRAM cells fail to support the transpose read operations, which are essential to enable the back-propagation (BP) in a CIM macro. Several customized SRAM cell structures were proposed to enable transpose read [15], [16], [17], albeit at the cost of additional area cost, as shown in Fig. 1(b). In addition, the update of weights incurs remarkable energy costs. Because the capacity of CIM macro is often limited [9], [18], it is difficult to store all the DNN parameters in the CIM array at one time. New weight patterns are periodically written into the synapse cells to participate the in-memory computing, and the partial results will be merged to generate the actual outputs of each NN layer. The energy overheads and latency costs brought by the update process

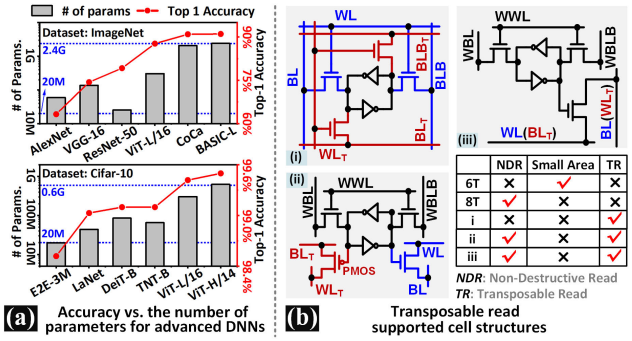

Fig. 1. (a) Top-1 accuracy versus the number of parameters for advanced DNNs on ImageNet and Cifar-10 datasets. (b) The transposable read supported cell structures: i, ii, and iii.

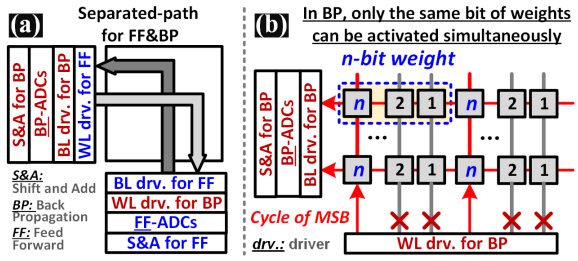

Fig. 2. On-chip training CIM engines suffer from (a) high cost of peripheral circuits for separate FF and BP paths and (b) a mapping mismatch problem in BP operation.

are rarely considered but harm EE and performance. Besides, to support both feed-forward (FF) and BP paths with CIM, traditional on-chip training accelerators usually require two separate sets of peripheral circuits to realize the normal and transpose read, as shown in Fig. 2(a) [4], [19], [20].

The conventional weight mapping method leads to limited parallelism and hardware utilization in the BP process for on-chip training with CIM. In conventional weight mapping, the 3-D convolutional (Conv) kernels are first flattened into 1-D column vectors, corresponding to different output channels. Then, the 1-D vectors with high-precision quantized weights are mapped to several adjacent columns of low-precision memory cells. For instance, in a 64-column array, eight kernels (output channels) with 8-bit precision can be mapped. This parallel mapping technique brings significant throughput during the execution of the FF process, whereas when performing the BP process with the same mapping method, a mapping mismatch problem arises. Specifically, the results of multiplying 1-bit gradients with different bits of weights on the same read word-line (RWL) cannot be directly added in the analog domain without proper shifting, or a MAC operation error will result. The reason lies in that a conventional mapping method used in the CIM array is mainly designed for inference tasks and lacks consideration for effective shifting required to support the BP process. To overcome the mapping mismatch problem, a common method is to ensure that only read bit-lines (RBLs) belonging to the same significant bit of weights are activated during a computation cycle, as shown in Fig. 2(b). However, in a CIM array storing n-bit weights, a maximum $1/n$ of all columns

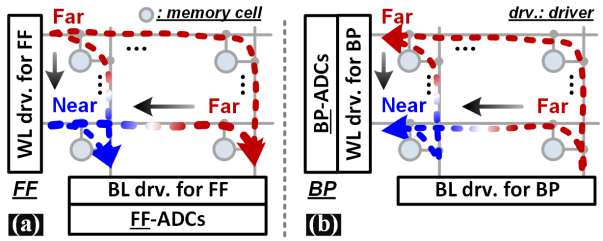

Fig. 3. IR-drop variation affecting MAC accuracy in current-sum CIMs during (a) FF and (b) BP processes.

can participate in the calculation at the same time, resulting in much less input parallelism in the BP path than that for FF.

The effect of IR drop variation deteriorates the accuracy of analog MVM outputs when numerous BLs or WLs are activated in parallel [21]. Published CIM accelerators usually employ an array capacity of approximately 1Mb, with tens to hundreds of rows or columns activated in a computation cycle [22], [23]. Increasing the parallel activation of rows or columns enhances the throughput but can lead to excessive readout current on a single RBL. Additionally, the compact SRAM array layout with long and narrow RBL metal lines brings considerable parasitic resistance. The large readout currents and nonnegligible parasitic resistances bring significant voltage discrepancies between the far-end SRAM cells and the near-end ones on the same load line, severely degrading the computation accuracy, as shown in Fig. 3. Furthermore, the IR drop variation tends to get worse at the advanced process nodes [24]. Besides, activating too many rows/columns simultaneously will result in a diminished output signal margin given the limited dynamic voltage range, which further calls for high-precision ADCs and leaves the output signal vulnerable to noise disturbances [2], [25], [26]. Therefore, the errors introduced by the IR drop variation effect hinder the parallelism in CIM operations.

In this work, a 28-nm 36 Kb SRAM CIM engine with NDR cell and weight update energy saving for MAC acceleration in AI inference and train applications. A 4T1T SRAM bit-cell is proposed with NDR and records the smallest cell size of 0.173 μm^2. The power-ON self-load-0 feature of the 4T1T cell saves the weight update energy and latency for writing 0. The shared-path dual-mode read (SPDMR) brings fewer circuit overheads to support both FF and BP paths. The bit-interleaving weight mapping (BIWM) speeds up the BP path without slowing FF. IR-drop-aware adaptive clampers (IRDAA-Cs) with hierarchical RWLs and read bit-lines (RBLs) apply possibly accurate voltages on near/far cells. The evaluation verified the proposed techniques.

The remaining sections of this article are structured as follows: Section II presents the 4T1T SRAM cell. Circuit techniques for arrays and peripherals are discussed in Section III. The evaluation is presented in Section IV. Finally, conclusions are provided in Section V.

II. 4T1T SRAM Cell

A. Proposed 4T1T SRAM Cell Structure

The novel 4T1T SRAM cell is proposed to address the first challenge of high-weight storage and update costs, as shown

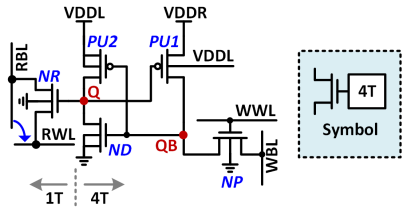

Fig. 4. The schematic of the proposed 4T1T NDR SRAM cell.

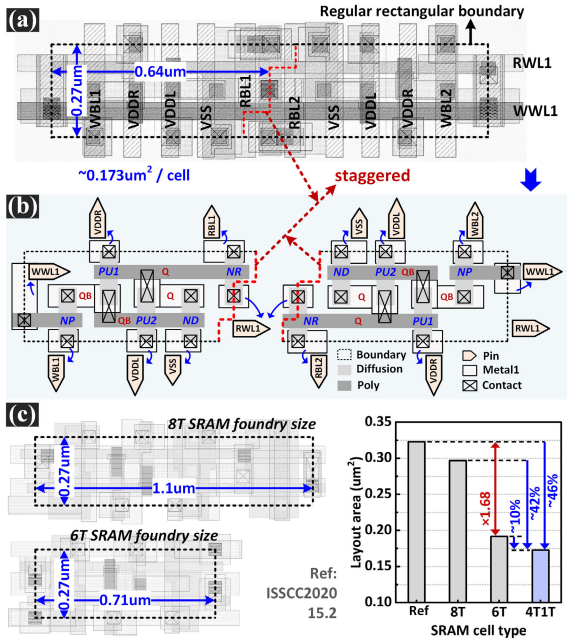

Fig. 5. Cell layout. (a) Basic layout of two 4T1T SRAM cells with regular rectangular boundaries. (b) Detailed diagram of the basic 4T1T layout. (c) Comparison with published or foundry-supplied layouts of SRAM cells.

in Fig. 4(a).This cell comprises two pull-up transistors (PU1, PU2), a pull-down transistor (ND), a read transistor (NR), and a write transistor (NP). The data are held in the storage nodes Q and QB of the asymmetric latch formed by PU1, PU2, and ND. The write WL (WWL) and write BL (WBL) are connected to the gate and drain of the write transistor NP, and RWL and RBL are connected to the source and drain of the read transistor NR, respectively. The sources of PU1 and PU2 are connected to separate power supplies, VDDR and VDDL, respectively, while their substrates are jointly connected to VDDL. The substrates of the three NMOSs, NR, ND, and NP, are connected to the ground.

The layout of the 4T1T SRAM cell is shown in Fig. 5. Two 4T1T cells form a rectangular shape with a staggered boundary to save area. In the 28 nm process, each 4T1T cell has an average area of ~0.173 μm^2, as shown in Fig. 5(a). The specific details of 4T1T SRAM cell layout are shown in Fig. 5(b), where the wiring information of each signal and transistor is clearly marked. The power and ground of the 4T1T cell are aligned vertically, similar to the 8T SRAM cell layout, except for separate alignment of VDDL and VDDR. Thanks to the decoupled read path from the internal storage nodes, the proposed 4T1T cell features NDR and only occupies a similar placement to the 6T. The contact vias of

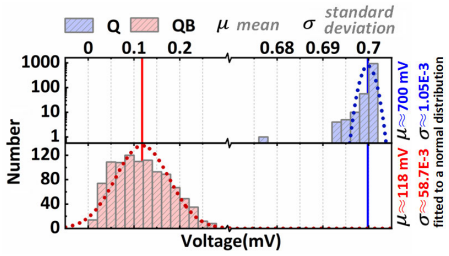

Fig. 6. Results of 1k-point MC simulation for long-term maintenance (up to 10^9 seconds) of QB node at 0 in the standby state of the 4T1T cell.

RBL1 and RBL2 are centrosymmetric with respect to the contact vias of RWL1, ensuring balanced read path parasitism for the two cells in the same basic layout. This design avoids cell-level parasitic inconsistency caused by asymmetric layout and simplifies the analysis of CIM operation with IR drop problem in hardware. Compared with the published SRAM cell for on-chip training [18], the 8T and 6T SRAM cells provided by the foundry, the proposed 4T1T SRAM cell is 46%, 42%, and 10% smaller, respectively, as illustrated in Fig. 5(c). The proposed 4T1T bit-cell and the SRAM bit-cells used for comparison are all based on push-rule.

The removal of one pull-down transistor in the 4T1T SRAM cell may cause the QB node to store a weak "0" that is easily lost due to leakage from PU1. To ensure the stability of the QB node to store 0, it can be realized by setting VDDL at a higher voltage level than VDDR. Meanwhile, both the WWL and WBL are pulled down to the ground during the standby or read state, and the subthreshold leakage of NP contributes to maintaining QB remains at 0. In this work, VDDR and VDDL are set as 0.5 and 0.7 V, respectively. Therefore, the source-substrate voltage difference (V_{SB}) of PU1 is greater than that of PU2. In this way, the threshold voltage (V_{TH}) of PU1 is higher than that of PU2 according to the body effect. The subthreshold conductivity of PU1 is suppressed by the increasing of V_{TH}, indicating that PU1 has less subthreshold leakage to the QB node when QB = 0, that is, QB is more difficult to be pulled up by PU1 and thus QB maintained at a stable low voltage level. In addition, when Q = 1, the gate of PU1 is subject to a higher voltage than its source, promoting a better turn-off of the PU1 and further suppressing the subthreshold conductivity of PU1. To verify the stability of the QB node to store 0, the 1k-point Monte Carlo (MC) simulation of the 4T1T SRAM cell retention is conducted, and the results are shown in Fig. 6. It was observed that when the 4T1T SRAM cell in the standby state stores data 1 (Q = 1 and QB = 0) for a duration of up to 10^9 s, the average voltage at the QB node remains around 118 mV, which allows the reliable pulling up of the Q node by PU2. To evaluate the leakage of the 4T1T bit-cell, a 1k-point MC simulation is conducted. When Q = 0, the average leakage power of a 4T1T cell is ~ 1.93 nW, primarily due to the nonzero gate–source voltage of PU2. When Q = 1, the average leakage power of a 4T1T cell is ~ 1.26nW, mainly attributed to the nonzero gate voltage of ND. The leakage current of the 4T1T bit-cell array is much smaller than that of the peripheral circuits.

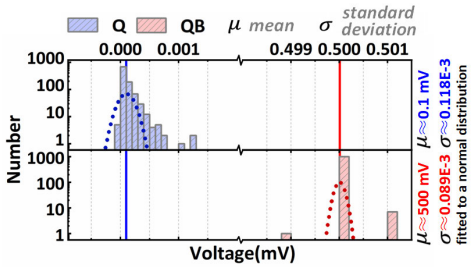

Fig. 7. Results of 1k-point MC simulation for self-loading of Q node to data 0 after power-ON.

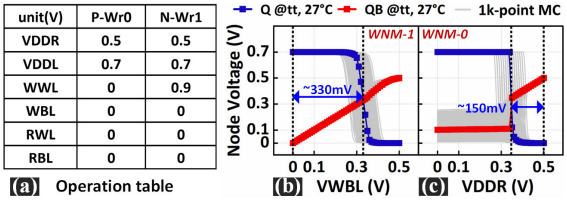

Fig. 8. (a) Write operation table. The WNM of (b) normal write 1 (WNM-1), and (c) power-ON self-load-0 (WNM-0).

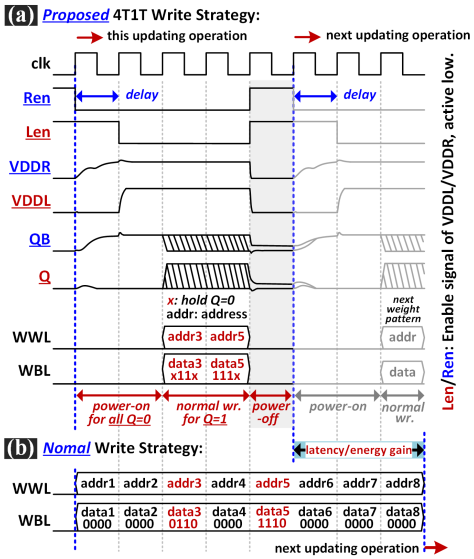

Fig. 9. Time diagram of (a) proposed 4T1T write strategy and (b) normal write strategy.

B. Write Operations

The proposed 4T1T SRAM cell entails two forms of write operations: the normal write 1 to Q node operation and the power-ON self-load-0 to Q node operation. For the normal write 1 to Q node operation, the WWL is set to 0.9 V, and the WBL is pulled down to the ground. The power-ON self-load-0 to Q node operation upon power-ON is a new feature of the 4T1T cell owing to the removal of the pull-down transistor on the QB side. In detail, due to the presence of a positive feedback loop formed by PU1 and ND, the QB (Q) node tends to transition to a high (low) voltage level during power-on. The enable signal Len (Ren) is utilized to control the power-up process of VDDL (VDDR). When Ren is activated (set to 0) one clock cycle earlier than Len, VDDR initially exceeds VDDL. During this period, the threshold voltage of PU1 becomes smaller than PU2, causing QB to bias to a higher voltage (approximately 0.5 V) before Q, thereby forming a positive feedback loop with QB = 1 and Q = 0. Even as VDDL gradually stabilizes at 0.7 V, higher than VDDR, this positive feedback loop remains undisturbed. Certain storage media exhibit the condition of being set to 0 upon power-up, like eDRAM. Note that the purported power-up zeroing in eDRAM relies on passive write-ins from peripheral circuits, which differ significantly from the proposed proactive self-load-0 operation in our work. Given the high sparsity commonly observed in DNNs [27], [28], this self-load-0 feature of 4T1T cells can be exploited. The 1k-point MC results of the voltages of Q and QB nodes after power-ON are shown in Fig. 7, demonstrating the stable self-load-0 function of the 4T1T cell.

The operation table of the normal write 1 to Q node operation and the power-on self-load-0 to Q node operation are summarized in Fig. 8(a). The write noise margin (WNM) for normal write 1 (WNM-1) and power-on self-load-0 (WNM-0) in the 4T1T SRAM cell are depicted in Fig. 8(b) and (c), acquired by a 1k-point MC simulation. The measurement of WNM-1 used the method of BL voltage analysis, wherein a dc input voltage is applied to the WBL to analyze the flipping voltage of the storage node [29]. The measured result is about 330 mV. Evaluating the WNM-0 involves fixing VDDL while VDDR transitions from a high level to ground. The initial states of Q and QB are set to 0 and 1, respectively, and the voltage of VDDR is observed when QB flips from 1 to 0. The flip voltage of VDDR, known as WNM-0, signifies the range in which the QB node remains stable when writing 0. The average of WNM-0 is about 150 mV. As for the impact of continuous write operations on the data stored in bit-cells, we find that row-by-row weight updates do not cause fatal errors to Q/QB node.

Based on the above two write operation modes, this work proposes an efficient write strategy for weight updating, called 4T1T write strategy, which includes three states: power ON (State-1), normal write (State-2), and power OFF (State-3), as shown in Fig. 9(a). In State-1, the power-on self-load-0 to Q node operation feature of the 4T1T bit-cell is utilized by activating the corresponding power control module of VDDR one clock cycle ahead of VDDL. During State-2, only the non-all-zero rows within the weight pattern need to be written to the 4T1T array. Unlike the normal write strategy depicted in Fig. 9(b), the 4T1T write strategy allows for the omission of all-zero rows to obtain energy and latency savings. Following State-2, the SRAM array executes the CIM operation necessary for the FF or BP algorithm, which is not discussed in Fig. 9. After executing the CIM operation and obtaining a new weight pattern from the on-chip buffer, the 4T1T array transitions to State-3 to clear the data stored in SRAM cells. This is achieved by pulling both VDDL and VDDR down to the ground, at which point all Q and QB nodes are set to 0.

Work [30] published a 5T SRAM bit-cell structure, which also removes one side of the pull-down path. However, it does not implement the self-load-0 feature. Meanwhile, the read path of this 5T cell follows the conventional transfer gate

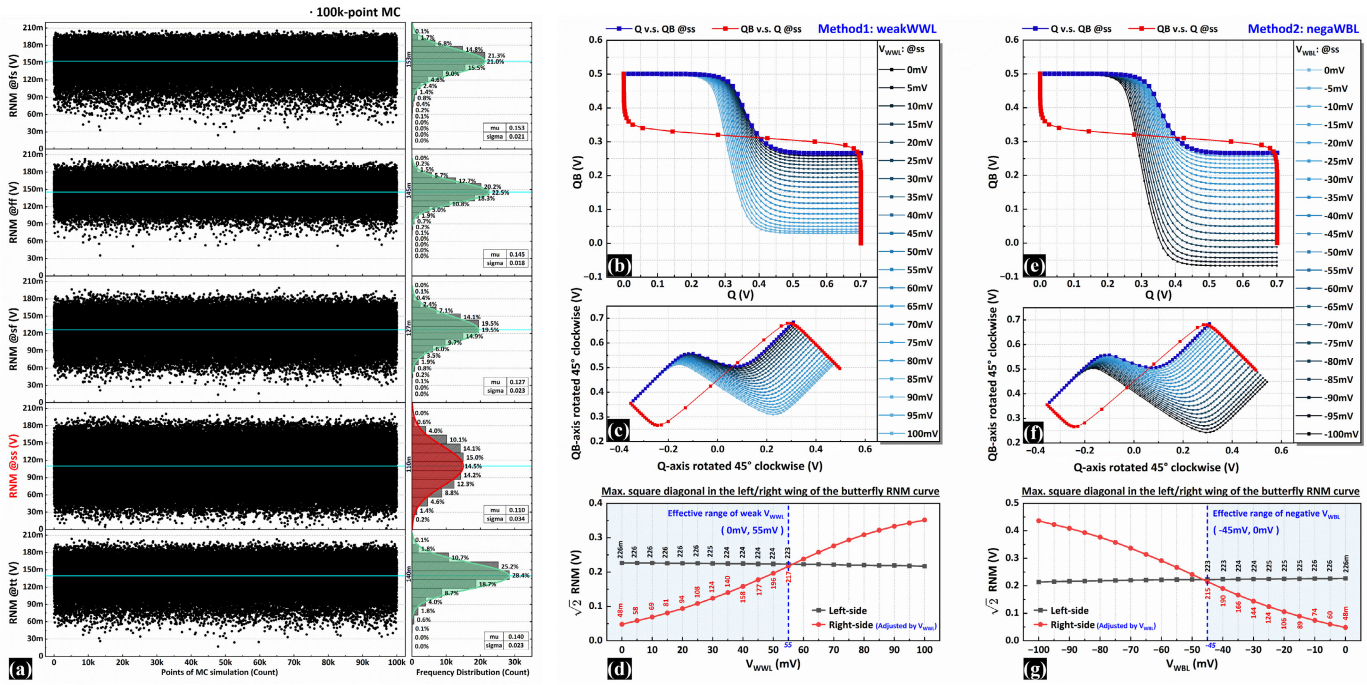

Fig. 10. (a) 100k-point MC simulation results of the 4T1T bit-cell RNM with local variation @$tt/ss/ff/sf/fs$ global corners. Comparison of the "weakWWL" and "negaWBL" methods for 4T1T RNM optimization: (b) RNM butterfly curve with V_{WWL} swept from 0 to 100 mV in 5 mV steps; (c) RNM curve with 45° clockwise rotation; (d) maximum embedded square diagonals (i.e., $\sqrt{2}$RNM) of the right/left wing of the butterfly curve for different V_{WWL} values; (e) RNM butterfly curve with V_{WBL} swept from −100 to 0 mV in 5 mV steps; (f) rotated RNM curve; and (g) the $\sqrt{2}$RNM for different V_{WBL} values.

structure found in 6T SRAM, leading to intrusive readout at the Q node. In addition, this 5T cell designs the write path as a low-threshold transistor and biases BLB to always ground except when writing a "1," resulting in significant static power consumption. Moreover, the layout of this 5T cell, unlike the staggered layout of our 4T1T SRAM, follows a conventional rectangular pattern. As a result, the area cost of the 5T cell is equivalent to that of 6T SRAM.

C. Read and CIM Operations

For the read operation, the single-ended read of the 4T1T cell is achieved by simply applying voltages to the RBL and RWL, and sensing the current through the channel of NR. Meanwhile, by decoupling the read path from the internal storage node Q, the 4T1T cell enables the NDR-like 8T cell, while only occupying a similar area of the 6T cell. The 4T1T cell supports two forms of NDR operations: normal and transpose read, which allows the input to be injected from the horizontal RWL and read out from the vertical RBL, or vice versa.

The transpose read operation provides the 4T1T SRAM array with hardware support for the implementation of parallel MVM after transposition, which constitutes the core operation of the BP process for on-chip training. Consequently, in the 4T1T-based CIM engine, the hardware execution of FF and BP processes can utilize the same weight pattern without the need for data transposition and subsequent storage, thereby reducing the complexity of top-level design and saving hardware resources. The read noise margin (RNM) of the 4T1T cell is measured using the voltage transfer characteristic butterfly curve method [31]. Note that because the 4T1T SRAM cell possesses the characteristics of NDR, its standby noise margin is essentially equivalent to the RNM. Fig. 10(a) presents the results of the 100k-point MC simulations @$tt/ff/ss/fs/sf$ five global corners with local variations. All the RNMs @$tt/ff/fs/sf$ global corners show relatively acceptable distribution, and only <0.06% of the iterations have RNM < 15 mV @ss global corner. To be precise, it is 58 out of the 100 000-point results. To optimize the RNM @ss, we provide two methods: weakly activating WWL below the threshold (Method1: "weakWWL") and driving WBL to a negative voltage (Method2: "negaWBL"). Fig. 10(b)–(g) compares the effect of the two methods on the 4T1T RNM @ss. For weakWWL (negaWBL), the butterfly curves and corresponding RNM can be obtained by sweeping V_{WWL}(V_{WBL}) in the ranges of 0100 mV (−100 to 0 mV) with a step size of 5 mV. Compared to the original RNM of the 4T1T cell @ss, the optimized RNM using weakWWL (negaWBL) method can achieve a 4.5× improvement at $V_{\mathrm{WWL}} = 55$ mV ($V_{\mathrm{WBL}} = -45$ mV) at the most. When Q = 0/1, the maximum WBL leakage is 7.2/3.7 pA for weakWWL ($V_{\mathrm{WWL}} = 55$ mV) and 6.7/3.7 pA for negaWBL ($V_{\mathrm{WBL}} = -45$ mV), which is really negligible.

Regarding the CIM operation for on-chip training, it can be viewed as a read operation with multiple rows and columns activated in parallel. Unlike the standard read operation, additional digital-to-analog converters (DACs) are required for RWLs/RBLs driving during the FF/BP process, and the quantization of analog outputs on RBLs/RWLs requires higher-precision ADCs other than 1-bit SAs. Generally, conventional CIM engines used for on-chip training employ separate peripheral circuits for the FF and BP paths, which

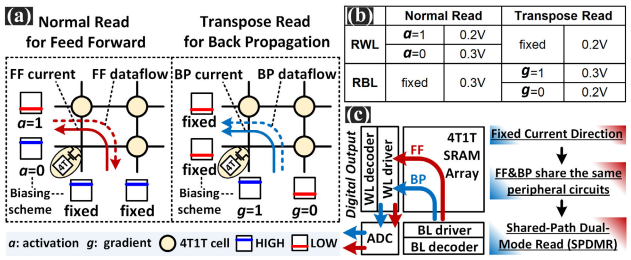

Fig. 11. (a) Schematic, (b) biasing table, and (c) illustration of SPDMR scheme.

result in the duplication of ADCs, driver circuits, and shift accumulators, adding to the circuit cost. Such configurations are not suitable for resource-constrained edge devices [17], [18].

This work proposes an efficient read strategy for on-chip training CIM engine, called the SPDMR scheme, wherein the normal read for FF and the transpose read for BP share the same path and the same physical current direction, enabling them to share a common set of peripheral circuits. Specifically, assuming a 4T1T cell with a Q node storing data 1, the read channel is activated. In normal (transpose) read, the RBLs (RWLs) are fixed to be HIGH (LOW), and the RWLs (RBLs) are clamped to be LOW (HIGH) for activation (gradient) of 1, or HIGH (LOW) for 0, as shown in Fig. 11(a). With SPDMR, an input can be applied to the RWL (RBL) to represent the activation (gradient) without storing a new transpose weight matrix, while unifying the array output currents to the same direction for both FF and BP path, that is, flowing from RBL to RWL. The biasing table for SPDMR is depicted in Fig. 11(b). We choose 0.3 and 0.2 V as the HIGH and LOW read voltages, respectively. In terms of the peripheral circuits, the array outputs during both FF and BP operations represent signals of the same physical quantity with consistent direction but varying magnitudes. Therefore, they can be uniformly processed, as illustrated in Fig. 11(c). Hence, there is no need to differentiate between the two paths, and they can be regarded as a kind of CIM operation, employing the same peripheral circuits for driving, sampling, holding, and quantifying, lowering the costs of the on-chip training engine. In our design, the on-chip implementation of the data path starts from the input of the activations to the ends of the quantized output of the ADC.

III. Circuit Techniques for CIM Engine

A. BIWM Scheme

We start by defining key concepts and abbreviations used throughout this article. The precision of signed activation, gradient, and weight is **m**-bit, **k**-bit, and **n**-bit, and they are abbreviated as **a**, **g**, and **w**, respectively. The convolution kernel size is represented as $\mathbf{t} \times \mathbf{t}$ in this investigation. The number of input and output channels of a convolution layer (ConvL) is denoted as $\mathbf{in_{ch}}$ and $\mathbf{ou_{ch}}$, respectively.

To fix the mapping mismatch problem discussed in Section I, a BIWM scheme is adopted in this work, which involves three steps as illustrated in Fig. 12. The first step is kernel flattening, where the 3-D kernel matrix is expanded into

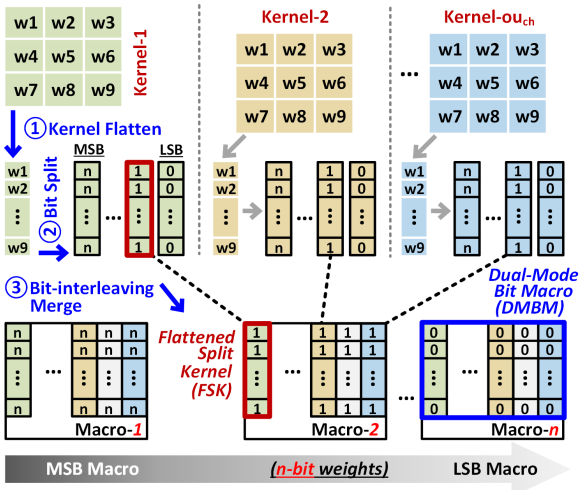

Fig. 12. Illustration of the BIWM scheme with 3 × 3 kernels and n-bit weights.

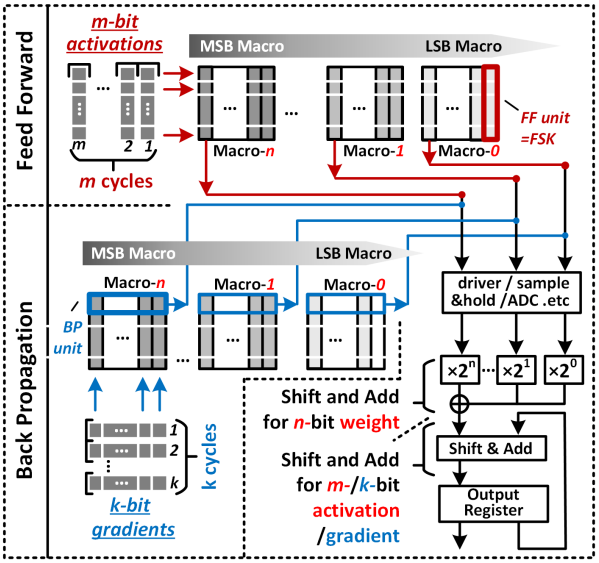

Fig. 13. Dataflow of FF/BP algorithm in a bit-interleaving fashion.

a 1-D column vector. The second step is bit splitting, where the quantized n-bit column vector is split into n binary column vectors known as flattened split kernels. The third step is bit-interleaving merge, where the flattened split kernels of the same bit from different kernels are merged and mapped onto the same dual-mode bit macro (DMBM). As a result, binary 2-D DMBMs with the number of n and the size of $(t \times t \times in_{ch})$-row$\times ou_{ch}$-column are obtained from the n-bit 3-D weight matrix with ou_{ch} number and t-row $\times t$-column$\times in_{ch}$-depth size after applying the BIWM scheme. Note that all columns in a DMBM can be read simultaneously in each computation cycle.

Since the BIWM scheme alters the mapping rules within the array, the peripheral circuits of the array are modified accordingly, as shown in Fig. 13. Outputs from the same DMBM exhibit the same shift scale. For example, the outputs of the DMBM responsible for mapping the MSB (LSB) of n-bit kernels are quantized by the ADCs and uniformly shifted

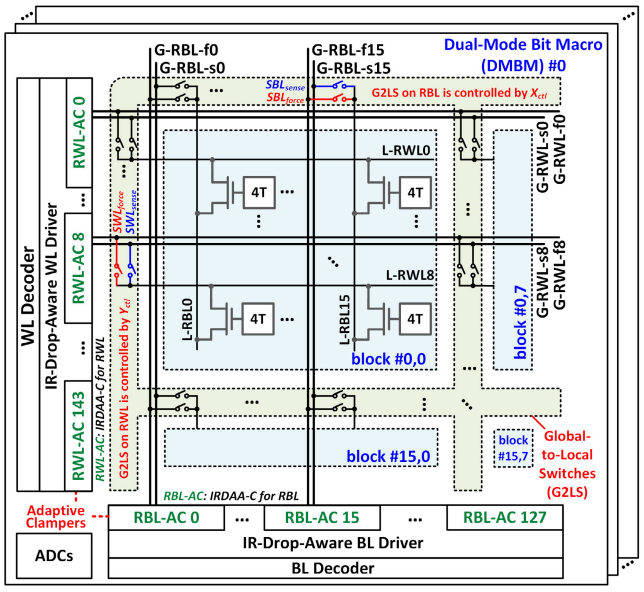

Fig. 14. IR-drop-aware architecture of the proposed CIM engine.

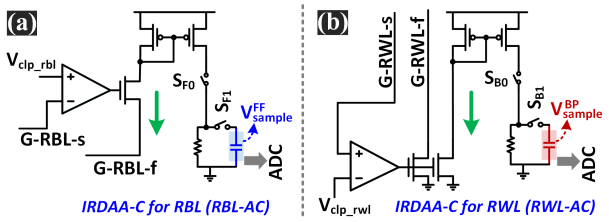

Fig. 15. Schematic of IR-drop-aware adaptive clampers (IRDAA-Cs). (a) IRDAA-C for RBL (RBL-AC). (b) IRDAA-C for RWL (RWL-AC).

left by n-bits (not shifted). The quantized outputs of DMBMs belonging to different bits but the same weight kernel will be accumulated after bit-shift to generate the MAC results of 1-bit activations (gradients) and n-bit weights for the FF (BP) path. For an MVM operation of an (a) m-bit (k-bit) activation (gradient) vector with an n-bit weight (transposed weight) matrix, the above procedure is repeated m (k) times, followed by a shift-accumulation based on the significance of the activation (gradient) bits. The final output of this ConvL is stored in output registers for on-chip training in the subsequent ConvL. Each DMBM contains weight data with the same significant bit, thus eliminating the mapping mismatch problem within the same RWL.

B. IRDAA Scheme

In this article, we propose an IRDAA scheme suitable for conventional DNN mappings to mitigate the IR drop problem in advanced process nodes. We denote local (global) RWL and local (global) RBL as L-RWL (G-RWL) and L-RBL (G-RBL) for brevity. The terms "force" and "sense" are abbreviated as "**-f**" and "**-s**," respectively, in reference to Kelvin test structures which represent the current and potential lines [32], [33]. The Kelvin test structure, developed as a resistance measurement technique to address contact resistance impact and ensure precise measurements, involves "force" and "sense" components dedicated to current application and voltage measurement, respectively.

First, an SRAM CIM array, or called DMBM, is divided into multiple equally-sized smaller blocks by differentiating local and global portions of RBL and RWL, as shown in Fig. 14. Each row (column) segment of a block, consists of a group of 16 (9) 4T1T cells, which share the same L-RWL (L-RBL), as shown in Fig. 14. The rationale behind setting the input parallelism to 9 for the FF mode is rooted in the prevalence of a 3×3 convolutional kernel size in DNNs. Larger kernel sizes, such as 5×5 or 7×7, can be effectively represented through the aggregation of multiple layers of 3×3 kernels [34]. The choice of an input parallelism of 16 for the BP mode stems from the common practice of defining the output channel quantity in convolutional layers as powers of 2.

The complete rows (columns) of the CIM array are composed of 8 (16) distinct blocks. L-RWL (L-RBL) connections in multiple blocks in the same row (column) are linked to a G-RWL (G-RBL) through the global-to-local switches (G2LSs). Each block is equipped with four dedicated G2LSs, namely $\text{SBL}_{\text{force}}$ and $\text{SBL}_{\text{sense}}$, responsible for connecting L-RBL to G-RBL-f and G-RBL-s, controlled by the same gate control signal $\mathbf{X_{ctl}}$; and $\text{SWL}_{\text{force}}$ and $\text{SWL}_{\text{sense}}$, responsible for connecting L-RWL to G-RWL-f and G-RWL-s, controlled by another gate control signal $\mathbf{Y_{ctl}}$. Each block can be selected and accessed by its specific $[X_{\text{ctl}}, Y_{\text{ctl}}]$ combination through the global interconnects. The blocks in the same column (row) share the same X_{ctl} (Y_{ctl}), thus requiring 8 (16) X_{ctl} (Y_{ctl}) signals in total for a DMBM, designated as $X_{\text{ctl0}}X_{\text{ctl7}}$ ($Y_{\text{ctl0}}Y_{\text{ctl15}}$). During the operation of the FF (BP) path, 8 (16) blocks located in the same row (column) are activated for parallel execution of MVM (transpose MVM) in one computation cycle. At this stage, Y_{ctl} (X_{ctl}) adopts a one-hot code to enable one specific row (column) of blocks within the DMBM, while all X_{ctl} (Y_{ctl}) signals are active to enable all columns (rows) of blocks in that row (column). Therefore, in the FF (BP) mode, the computational parallelism within a DMBM is 9-row × 128-column (144-row × 16-column).

Second, the double-wiring G-RBLs (G-RWLs) are configured with separate force and sense channels, as shown in Fig. 15. The force channel is used for applying voltages to the selected RBL (RWL), while the sense channel, which carries no direct current, accurately measures the voltage applied to the selected L-RBL (L-RWL). The IR-drop aware adaptive clampers (IRDAA-C) play a vital role in the IRDAA scheme by generating large enough voltages with IR-drop considered according to the feedback result from the sense channel to provide dedicated biasing voltages. Note that the circuit involved in the IRDAA scheme does not include ADCs.

During the FF process, the output current is read from the G-RBL-f. Considering that the current flows from RBL to RWL, a primary current mirror composed of PMOS transistors is employed to achieve proportional replication of the output current. When the switch SF_0 is active, the output current is converted to voltage and sampled on the resistor. When the switch SF_1 is active, the stable output voltage on the sampling resistor is buffered by a capacitor to await ADC quantization, as shown in Fig. 15(a). As for the BP process, the output

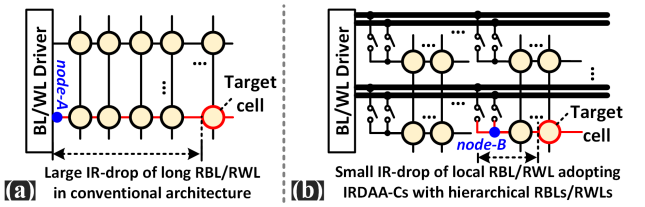

Fig. 16. Equivalent circuits for IR-drop analysis: (a) without or (b) with hierarchical RBLs/RWLs.

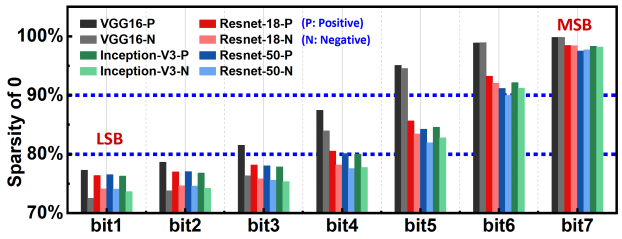

Fig. 17. Bitwise sparsity of VGG16, Resnet-18, Resnet-50, and Inception-V3.

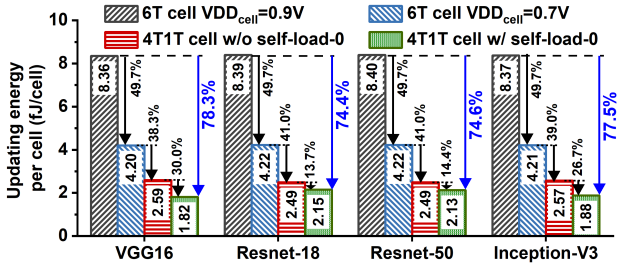

Fig. 18. Analysis of energy consumption reduction in write operations by the proposed 4T1T cell and 4T1T write strategy.

current is read from the G-RWL-f. As the current direction still flows from RBL to RWL, a two-stage current mirror is used to replicate the output current, as illustrated in Fig. 15(b). The first-stage current mirror consists of an NMOS transistor, with its size matched to the NMOS transistor connected to the output of the operational amplifier to form a negative feedback loop. Due to their gate connections to the same node with identical voltages, the drain currents of two NMOS transistors with the same width-to-length ratio are equal, enabling equal proportional replication of the output current on G-RWL-f. The second-stage current mirror and subsequent sample-and-hold circuit remain consistent with Fig. 15(a), which can be designed as a shared circuit during the FF and BP mapping to optimize circuit cost. In addition, the ADCs are shared between the FF and BP stages.

The IR drop mitigation scheme proposed in [35] uses a "pitch-matched read channel" structure to clamp the voltage between bitline and source-line, which also utilizes a combination of sense-force structure to enhance clamping effects. However, the proposed IRDAA scheme is more refined. We provide a set of clamping circuits for RWL and RBL, respectively, with the aim of clamping the voltages at RWL and RBL to fixed values. Moreover, we have also incorporated the G2LS structure, corresponding to the design of local/global RWL and local/global RBL.

Thanks to the hierarchical RWL and RBL design, combined with the proposed force/sense structure, the negative feedback node of the voltage follower has shifted from the boundary of the large CIM array (**node-A**) to the boundary of each small block (**node-B**), as shown in Fig. 16. The feedback node B has moved to a lower circuit level, which implies the use of lower-level metal wiring, fewer vias, shorter interconnect distances, and fewer driven SRAM cells are involved. These factors contribute to mitigating the circuit parasitic effects, suppressing the influence of IR drop, and obtaining more stable and accurate clamping voltages, thereby yielding more precise CIM results.

IV. Evaluation

A. Performance of Proposed Schemes

Our work involves the design at the 28 nm node, where the selection of various analog voltages, such as the read voltages for RWL (0.2 V) and RBL (0.3 V), as well as the supply voltages for VDDL (0.7 V) and VDDR (0.5 V), is based on extensive circuit simulation data. The choice of these voltages also considers analyses of various noise tolerances, circuit stability, and optimization for hardware efficiency. The final decisions are made through a comprehensive evaluation process.

In this part, we will discuss four commonly used DNN models: VGG16, Resnet-18, Resnet-50, and Incepation-V3. All used models are pretrained on the ImageNet-1K dataset and quantized into 8-bit by official PyTorch sources. To explore the benefits of the 4T1T write strategy in practical DNN updates, we conducted a statistical analysis of the bitwise sparsity of VGG16, ResNet-18, ResNet-50, and Inception-V3, as shown in Fig. 17. It was observed that the sparsity of zeros increases monotonically from the LSB to the MSB. This finding aligns with the software-side analysis of DNNs, indicating that the weights with larger magnitudes play more crucial roles and tend to have fewer occurrences [36]. Moreover, the lowest sparsity levels of the four DNNs, starting from the LSB, are all greater than 70%, which precisely corresponds to the range where the 4T1T write strategy exhibits significant effectiveness. This indicates that the common DNN parameters are rich in zeros, making them favorable for the proposed 4T1T write strategy. It implies that more write operations can be skipped after power-ON, leading to reduced energy consumption and latency in write updates.

Fig. 18 illustrates the simulated average updating energy per cell of the proposed 4T1T SRAM array (with 4T1T or normal write strategy) and a 6T SRAM array (with normal write strategy) when loading these four DNNs. It is observed that when maintaining the storage medium as 6T SRAM and applying only voltage scaling (VDD reduced from 0.9 to 0.7 V), an average reduction of approximately 49.7% in write operation energy consumption can be achieved. Furthermore, transitioning from 6T SRAM to 4T1T SRAM provides an additional average reduction of around 39.8%. Upon the foundation of this second optimization, further refining the write operation strategy from the normal write strategy to the 4T1T write strategy proposed in this article yields an additional average reduction of approximately 21.2% in write operation

TABLE I
LATENCY SAVINGS OF 4T1T WRITE STRATEGY

DNNs	VGG16	Resnet-18	Resnet-50	Inception-V3
Latency Saving	**39.1%**	**19.7%**	**20.5%**	**34.6%**

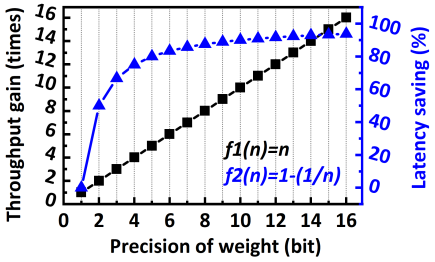

Fig. 19. Effectiveness of the BIWM scheme.

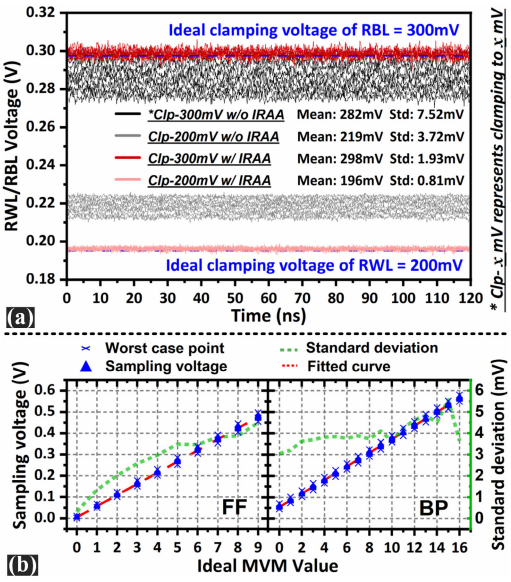

Fig. 20. Clamping effectiveness of the IRDAA scheme. (a) Comparison with the traditional scheme when considering circuit noise influence. (b) Linearity of all possible states of a block output during FF and BP execution.

energy consumption. Table I presents the simulated latency savings of 4T1T write strategy working on different DNN models, ranging from 19.7% to 39.1%. Note that the provided data on latency saving does not stem from the analysis of random data but is directly derived from the statistical analysis of weight data for the four DNNs mentioned above. Note that the DNN models can be optimized to be sparser or more structurally regular using pruning, clustering, and so on, before mapping onto the 4T1T CIM hardware for better performance.

Fig. 19 illustrates the throughput gain brought by the BIWM scheme in the BP process. For n-bit weights, BIWM can improve the throughput by up to n times, which indicates a $1-(1/n)$ latency saving for the BP process. Thanks to the distributed mapping of weight bits with different significance, the maximum parallelism in BP is not as restricted as conventional mapping, and the hardware resources can be efficiently utilized. Note that for small systems in which the number of CIM macros k is smaller than the weight precision,

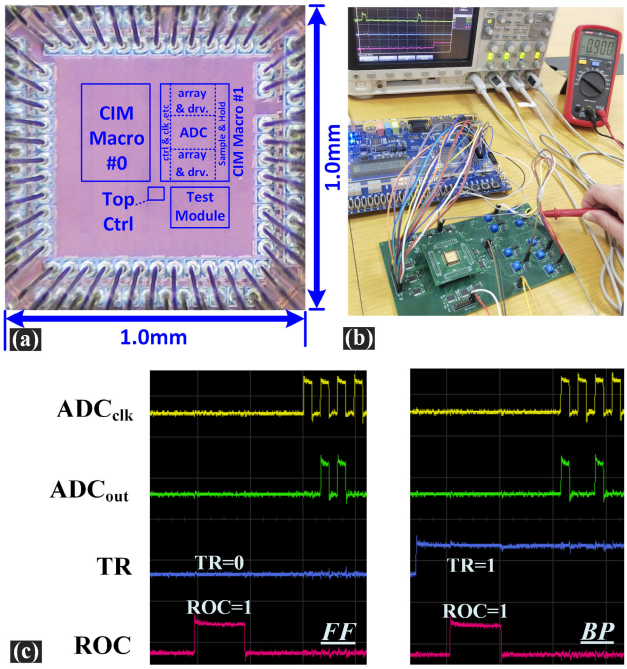

Fig. 21. Hardware Information. (a) Die photo. (b) Test setup. (c) Measurement results of MAC for FF/BP.

TABLE II
HARDWARE VERSUS SOFTWARE ACCURACY

Dataset (NN Style)	Cifar10 (VGG-8)	ImageNet (VGG-B)
Accuracy of Chip (vs Sim.)	90.23% (90.41%)	67.11% (67.23%)

BIWM cannot be fully executed, but still can improve the BP performance by k times.

Fig. 20(a)compares the simulated clamping voltages on RWL/RBLs with/without IRDAA in the 28 nm process. The IRDAA circuit structure, besides addressing the core objective of mitigating IR drop effects, also provides concurrent alleviation of noise and circuit variation. The conventional clamping scheme shows an average 18 mV (19 mV) deviation, which will consequently cause an over 37% error in the output sampling voltage. With the help of IRDAA, on the other side, only an average 2 mV (4 mV) deviation is shown. Besides, clamping without IRDAA has much worse stability for far and near cells, which could vary from 275 to 305 mV (210–225 mV) for 300 mV (200 mV) clamping, when applying different input and weight patterns. The adoption of IRDAA effectively restricted the clamping voltages to be rather close for all the cells to execute both accurately and stable in storage multiplication. Fig. 20(b) presents the simulated sampling voltages for different ideal MVM output values in FF/BP. It can be observed that there is no overlap between different readout current states in both the FF and BP processes. This indicates that our circuit has reserved sufficient noise margin for each possible operational output state. Note that there is a reduction in standard deviation becomes apparent under conditions of high parallelism. This behavior occurs because the current mirror approaches its operational limit, resulting

TABLE III
COMPARISON WITH PREVIOUS WORKS

	JSSC'19 [1]	JSSC'20 [2]	JSSC'22 [4]	ISSCC'22 [5]	VLSI'22 [8]	ISSCC'23 [10]	CICC'23 [12]	**This work**
Process	55nm	65nm	28nm	28nm	22nm	22nm	65nm	**28nm**
SRAM Cell	T8T SRAM	8T + 2T SRAM	6T + TWT SRAM	6T SRAM	6T + 3T SRAM	6T + 3T1C SRAM	6T SRAM	**4T1T SRAM**
Activation (a) (bit)	1,2,4	8	FF:2,4,8 BP:8	4,8	8	8	3	**FF/BP:1~8 scalable**
Weight (w) (bit)	2,5,5	1,2,4,8	4,8	4,8	8	8	3	**FF/BP:1~8 scalable**
ADC/TDC [*1] (bit)	5	8	5	7[*1]	8	7	8	**<5**
Size of macro (bit)	3.75Kb	576Kb	64Kb	128Kb	128Kb	128Kb	108Kb	**36Kb**
Area/Cell [*2] (μm^2)	**1.58× 6T**	**1.8 (1.8× 6T)**	**1.68× 6T**	**0.379 (1.6× 6T)**	**N/A**	**N/A**	**2.6**	**0.173 (0.9× 6T)**
On-chip training	No	No	**Yes**	No	No	No	No	**Yes**
Optimized for weight update	No	No	No	No	No	No	No	**Yes**
Weight update energy (fJ/bit)	N/A	**600**	N/A	N/A	N/A	N/A	N/A	**1.8** [*3]
EE [*4] (TOPS/W)	72~89.2 (a:1b/w:3b)	192 (a:1b/w:1b)	FF:56.1~61.1 (a:2b/w:4b) BP:14.1~15 (a:8b/w:4b)	84.5~112.6 (a:4b/w:4b)	15.5~32.2 (a:8b/w:8b)	16~21.4 (a:8b/w:8b)	95.4 (a:3b/w:3b)	**FF:263.1 (a:1b/w:1b) BP:412.1 (a:1b/w:1b)**
AE [*4] (TOPS/mm^2)	N/A	0.6 (a:1b/w:1b)	N/A	N/A	2.4~4 (a:8b/w:8b)	1.2~1.4 (a:8b/w:8b)	3.1 (a:3b/w:3b)	**FF:2.5 (a:1b/w:1b) BP:4.9 (a:1b/w:1b)**

[*1] TDC: Time-to-Digital Converter
[*2] 6T: Compact/Standard 6T SRAM bit-cell using foundry push rules.
[*3] Write update energy is based on 50% '0' and 50% '1' array data, simulated at the 28nm node.
[*4] Represented at the finest granularity the reported activation/weight can map.

in the suppression of output voltage along with the associated noise fluctuations.

B. Experiment Results

This work fabricated a 4T1T SRAM-based CIM test chip in a 28-nm CMOS process. Fig. 21(a) presents the die photo. This test chip consists of two 144 row × 128 column CIM macros (also called two CIM engines), a test macro, and a top control module. A CIM macro contains a 4T1T SRAM array made of 16 × 8 blocks, and each block comprises 9 row × 16 column 4T1T SRAM cells. The BIWM scheme demands multiple CIM macros for mapping to boost the BP parallelism, which means at least n CIM macros are needed for the weight precision of n-bit. On the test chip, due to the limited hardware resources, we realize BIWM mapping in a time-multiplexing way, letting each macro execute for different bits in different time periods.

Fig. 21(b) shows the photo of our test system. This chip is tested at 25 MHz. We designed an SAR ADC in the 28 nm process with a sampling frequency of 400 MS/s and a quantization precision of less than 5-bit. Since the ADC design is not the primary focus of this work, we referred to a previously published work [37]. This ADC is sufficient for activating one row of blocks (nine cells on each BL) for FF, or one column of blocks (16 cells on each BL) for BP in parallel. Note that the BIWM scheme allows the maximum parallelism for both FF and BP to be the complete array size, with ADCs of appropriate resolution. The 4T1T SRAM array works with VDDR at 0.5 V and VDDL at 0.7 V. WWL (WBL) is 0.9 V (0.5 V) when enabled during write operations. During the read or CIM operations, enabled RWL and RBL are 0.3 or 0.2 V according to the SPDMR scheme, ensuring the read current always flows from RBL to RWL. In our design, each macro is equipped with nine SAR ADCs. Thanks to the proposed SPDMR scheme, eight ADCs are effectively multiplexed for both FF and BP paths. Fig. 21(c) presents the captured output waveforms of MVM quantized outputs for FF and BP path, where TR is the control signal to enable FF (TR = 0) or BP (TR = 1), and ROC is the control signal to enable in-array multiplication (ROC = 1).

Considering nonlinearity and variation of the circuits [25], [38], [39], our chip achieves 90.23% and 67.11% (TOP1) accuracy for VGG-8 (CIFAR-10) and VGG-B (ImageNet),

respectively, as shown in Table II. Table III compares the proposed 4T1T SRAM-based CIM engine with the state of the art [1], [2], [3], [4], [6], [7], [8]. The 4T1T SRAM cell offers the capability of NDR like the 8T cell but with a 10% smaller cell area than the standard 6T cell. The 4T1T SRAM cell also supports transpose read operation, enabling a single CIM array to map the transposed data flows of FF/BP process. Combined with SPDMR, the engine allows for on-chip training while sharing ADCs, drivers, and other peripherals to achieve higher hardware efficiency [4]. Furthermore, the 4T1T cell supports self-loading of 0, making it suitable for mapping high-sparsity NNs. Compared to [2], the proposed work reduces the energy consumption for single-bit weight updates from 600 to 1.8 fJ, achieving a >90% optimization. Meanwhile, compared with traditional SRAM write schemes, the proposed updating method provides >70% power savings, as shown in Fig. 18. The BIWM scheme refines the hardware-mapping precision to 1-bit, supporting scalable activation/weight mapping. Lastly, the IRDAA-C scheme with hierarchical RWLs and RBLs effectively mitigates the impact of IR-drop on the accuracy of readout results within the CIM array. At the macro level, the proposed CIM engine achieves competitive EE of 263.1/412.1 TOPS/W and area efficiency (AE) of 2.5/4.9 TOPS/ mm^2 for FF/BP processes.

V. Conclusion

This work addresses some challenges associated with on-chip training applications in the CIM domain. A novel high-density 4T1T SRAM cell is first proposed with a compact size, nondestructive and transpose read capability, and power-on self-load-0 feature. The associated write strategy also enables energy-efficient and low-latency weight updates. Meanwhile, the SPDMR, BIWM, and IRDAA schemes are proposed for three distinct circuit-related issues, respectively: the high cost of peripheral circuits to support separate FF and BP paths, limited hardware parallelism in the BP process, and the notable impact of IR drop variation on the cell-level multiplication results. The evaluation from simulations and silicon data verified the proposed techniques.

References

[1] 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.

[2] 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, Sep. 2020.

[3] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," *ACM SIGARCH Comput. Archit. News*, vol. 23, no. 1, pp. 20–24, Mar. 1995.

[4] J.-W. Su et al., "Two-way transpose multibit 6T SRAM computing-in-memory macro for inference-training AI edge chips," *IEEE J. Solid-State Circuits*, vol. 57, no. 2, pp. 609–624, Feb. 2022.

[5] 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," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.

[6] S. Xie, C. Ni, A. Sayal, P. Jain, F. Hamzaoglu, and J. P. Kulkarni, "16.2 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*, vol. 64, Feb. 2021, pp. 248–250.

[7] P. Deaville, B. Zhang, and N. Verma, "A 22nm 128-kb MRAM row/column-parallel in-memory computing macro with memory-resistance boosting and multi-column ADC readout," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2022, pp. 268–269.

[8] P. Chen et al., "7.8 A 22nm delta-sigma computing-in-memory (ΔΣCIM) SRAM macro with near-zero-mean outputs and LSB-first ADCs achieving 21.38TOPS/W for 8b-MAC edge AI processing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 140–142.

[9] Y. Zhang et al., "Single-Mode CMOS 6T-SRAM macros with keeper-loading-free peripherals and row-separate dynamic body bias achieving 2.53fW/bit leakage for AIoT sensing platforms," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 184–185.

[10] H. Wang et al., "A 32.2 TOPS/W SRAM compute-in-memory macro employing a linear 8-bit C-2C ladder for charge domain computation in 22nm for edge inference," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2022, pp. 36–37.

[11] D. Kim, J. Kung, and S. Mukhopadhyay, "A power-aware digital multilayer perceptron accelerator with on-chip training based on approximate computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 2, pp. 164–178, Apr. 2017.

[12] Y.-J. Jo, B. P. Yap, D.-H. Yoon, H. Kim, Y. Zheng, and T. T. Kim, "DenseCIM: Binary weighted-capacitor SRAM computation-in-memory with column-by-column dynamic range calibration SAR ADC," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2023, pp. 1–2.

[13] S. Erera et al., "A summarization system for scientific documents," 2019, *arXiv:1908.11152*.

[14] A. Guler and N. K. Jha, "Three-Dimensional monolithic FinFET-based 8T SRAM cell design for enhanced read time and low leakage," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 4, pp. 899–912, Apr. 2019.

[15] J.-s. Seo et al., "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2011, pp. 1–4.

[16] K. Bong, S. Choi, C. Kim, S. Kang, Y. Kim, and H.-J. Yoo, "14.6 A 0.62mW ultra-low-power convolutional-neural-network face-recognition processor and a CIS integrated with always-on haar-like face detector," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 248–249.

[17] H. Jiang, X. Peng, S. Huang, and S. Yu, "CIMAT: A compute-in-memory architecture for on-chip training based on transpose SRAM arrays," *IEEE Trans. Comput.*, vol. 69, no. 7, pp. 944–954, Mar. 2020.

[18] J.-W. Su et al., "15.2 A 28nm 64Kb 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*, Feb. 2020, pp. 240–242.

[19] X. Peng, S. Huang, H. Jiang, A. Lu, and S. Yu, "DNN+NeuroSim V2.0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 11, pp. 2306–2319, Nov. 2021.

[20] W. Wan et al., "A compute-in-memory chip based on resistive random-access memory," *Nature*, vol. 608, no. 7923, pp. 504–512, Aug. 2022.

[21] B. Crafton, C. Talley, S. Spetalnick, J.-H. Yoon, and A. Raychowdhury, "Characterization and mitigation of IR-drop in RRAM-based compute in-memory," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2022, pp. 70–74.

[22] W.-S. Khwa et al., "A 40-nm, 2M-cell, 8b-precision, hybrid SLC-MLC PCM computing-in-memory macro with 20.5–65.0TOPS/W for tiny-Al edge devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 1–2.

[23] J.-M. Hung et al., "An 8-Mb DC-current-free binary-to-8b precision ReRAM nonvolatile computing-in-memory macro using time-space-readout with 1286.4-21.6TOPS/W for edge-AI devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 1–2.

[24] S.-H. Weng, Y.-M. Kuo, S.-C. Chang, and M. Marek-Sadowska, "Timing analysis considering IR drop waveforms in power gating designs," in *Proc. IEEE Int. Conf. Comput. Design*, Oct. 2008, pp. 532–537.

[25] C. Zhao, J. Fang, J. Jiang, X. Xue, and X. Zeng, "ARBiS: A hardware-efficient SRAM CIM CNN accelerator with cyclic-shift weight duplication and parasitic-capacitance charge sharing for AI edge application," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 1, pp. 364–377, Jan. 2023.

[26] S. Zhang, K. Huang, and H. Shen, "A robust 8-bit non-volatile computing-in-memory core for low-power parallel MAC operations," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 6, pp. 1867–1880, Jun. 2020.
[27] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–11.
[28] F. Liu et al., "SME: ReRAM-based sparse-multiplication-engine to squeeze-out bit sparsity of neural network," in *Proc. IEEE 39th Int. Conf. Comput. Design (ICCD)*, Oct. 2021, pp. 417–424.
[29] K. Zhang et al., "A 3-GHz 70-Mb SRAM in 65-nm CMOS technology with integrated column-based dynamic power supply," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 146–151, Jan. 2006.
[30] A. Teman, A. Mordakhay, J. Mezhibovsky, and A. Fish, "A 40-nm sub-threshold 5T SRAM bit cell with improved read and write stability," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 12, pp. 873–877, Dec. 2012.
[31] E. Seevinck, F. J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE J. Solid-State Circuits*, vol. SSC-22, no. 5, pp. 748–754, Oct. 1987.
[32] C. M. Mezzomo, M. Marin, C. Leyris, and G. Ghibaudo, "Mismatch measure improvement using Kelvin test structures in transistor pair configuration in sub-hundred nanometer MOSFET technology," in *Proc. IEEE Int. Conf. Microelectronic Test Struct.*, Mar. 2009, pp. 62–67.
[33] R. Kuroda et al., "Characterization of MOSFETs intrinsic performance using in-wafer advanced Kelvin-contact device structure for high performance CMOS LSIs," in *Proc. IEEE Int. Conf. Microelectronic Test Struct.*, Mar. 2008, pp. 155–159.
[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[35] S. D. Spetalnick et al., "A 2.38 MCells/mm^2 9.81 -350 TOPS/W RRAM compute-in-memory macro in 40nm CMOS with hybrid offset/IOFF cancellation and ICELL RBLSL drop mitigation," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2023, pp. 1–2.
[36] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.
[37] C. C. Liu, C. H. Kuo, and Y. Z. Lin, "A 10 bit 320 MS/s low-cost SAR ADC for IEEE 802.11 ac applications in 20 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 11, pp. 2645–2654, Sep. 2015.
[38] C. Zhao, J. Fang, J. Jiang, Z. Guo, X. Xue, and X. Zeng, "Intra-array Non-Idealities modeling and algorithm optimization for RRAM-based computing-in-memory applications," in *Proc. IEEE 14th Int. Conf. ASIC (ASICON)*, Oct. 2021, pp. 1–4.
[39] K. Zhou et al., "An energy efficient computing-in-memory accelerator with 1T2R cell and fully analog processing for edge AI applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 8, pp. 2932–2936, Aug. 2021.

Chenyang Zhao received the B.S. degree from the North University of China, Taiyuan, China, in 2019. She is currently pursuing the Ph.D. degree with the State Key Laboratory of ASIC and Systems, Fudan University, Shanghai, China.

Her current research interest is in the chip design of neural network accelerators based on computing-in-memory architecture.

Jinbei Fang received the B.S. degree from Fudan University, Shanghai, China, in 2020, where he is currently pursuing the Ph.D. degree with the State Key Laboratory of ASIC and Systems.

His current research interests focus on computing-in-memory neural network accelerators.

Xiaoli Huang received the B.S. and M.S. degrees in microelectronics from Fudan University, Shanghai, China, in 2020 and 2023, respectively.

Her current research interests include nonvolatile memory-based computing-in-memory circuits.

Deyang Chen received the B.S. and M.S. degrees in microelectronics from Fudan University, Shanghai, China, in 2020 and 2023, respectively.

His current research interests include nonvolatile memory-based computing-in-memory circuits for deep learning and mixed-signal circuit designs.

Zhiwang Guo received the B.S. degree from Xidian University, Xi'an, China, in 2020, where he is currently pursuing the Ph.D. degree with the State Key Laboratory of ASIC and Systems.

His current research interests focus on inference and training of computing-in-memory neural network accelerators.

Jingwen Jiang received the B.S. degree from Fudan University, Shanghai, China, in 2021, where she is currently pursuing the Ph.D. degree with the State Key Laboratory of ASIC and Systems.

Her current research interests include computing-in-memory architecture and neuromorphic circuits and systems.

Jiawei Wang received the M.S. degree in integrated circuit engineering from Ningbo University, Ningbo, China, in 2017. He is currently pursuing the Eng.D. degree in electrical information engineering with Fudan University, Shanghai, China.

His current research interests include analog and mixed-signal circuit design.

Jianguo Yang (Member, IEEE) received the Ph.D. degree in microelectronics from Fudan University, Shanghai, China, in 2016.

In 2016, he joined the Department of Microelectronics, Fudan University, as a Post-Doctoral Research Fellow. In 2019, he joined the Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China, as an Associate Professor. His research interests include memory circuit design, hardware security, and new computing paradigms.

Jun Han (Member, IEEE) received the B.S. degree from Xidian University, Xi'an, China, in 2000, and the Ph.D. degree in microelectronics from Fudan University, Shanghai, China, in 2006.

In July 2006, he joined Fudan University as an Assistant Professor, where he is currently a Full Professor with the State Key Laboratory of ASIC and Systems. He is working on a high-performance domain-specific processor, especially for digital signal processing and cryptography.

Peng Zhou (Senior Member, IEEE) received the bachelor's and Ph.D. degrees in physics from Fudan University, Shanghai, China, in 2000 and 2005, respectively.

He is currently a Full Professor of novel electronic devices and process with the School of Microelectronics, Fudan University. He won the Shanghai Youth Science and Technology Outstanding Contribution Award, the Ministry of Education, the Shanghai Natural Science Second Prize, and the NR45 Young Scientist Award. In the past five years, he published more than 100 papers in *Nature Nanotechnology*, *Nature Electronics*, *Nature Materials*, *Nature Communications*, IEDM, and others, with over 14 000 citations and 18 highly cited papers.

Xiaoyong Xue (Member, IEEE) received the Ph.D. degree in microelectronics from Fudan University, Shanghai, China, in 2011.

He joined the Department of Microelectronics, Fudan University, as a Post-Doctoral Research Fellow. He is currently an Associate Professor with Fudan University. His research interests include high-performance memory/storage, in-memory computing circuits, and systems.

Xiaoyang Zeng (Senior Member, IEEE) received a B.Sc. degree from Xiangtan University, Xiangtan, China, in 1996, and the Ph.D. degree (Hons.) from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Beijing, China, in 2001.

In 2001, he joined Fudan University, where he was a Post-Doctoral Researcher from March 2001 to February 2003. He has been with Fudan University as a faculty since 2003, where he is currently a Chair Professor and the Executing Director of the State Key Laboratory of ASIC and System. He has published more than 200 articles in such international journals and conferences as IEEE ISSCC, IEEE JOURNAL OF SOLID-STATE CIRCUITS, the IEEE TRANSACTIONS on CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, the IEEE TRANSACTIONS ON VERY LARGE-SCALE INTEGRATION (VLSI) SYSTEMS, the IEEE VLSI SYMPOSIA, IEEE CICC, IEEE ESSCIRC, IEEE ASP-DAC, and IEEE A-SSCC. He has applied for more than 120 patents. His research fields include information security chips, baseband processing technologies for wireless communication, mixed-signal IC designs, and ultralow power IC methodology.