SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2023, Vol. 66 182404:1–182404:11
https://doi.org/10.1007/s11432-022-3627-8

Design memristor-based computing-in-memory for AI accelerators considering the interplay between devices, circuits, and system

Junjie AN[1,2], Linfang WANG[2,4], Wang YE[2,4], Weizeng LI[2,4], Hanghang GAO[2,4], Zhi LI[2,4], Zhidao ZHOU[2,4], Jinghui TIAN[3], Jianfeng GAO[2], Chunmeng DOU[2,4*] & Qi LIU[3]

[1]*School of Microelectronics, University of Science and Technology of China, Hefei* 230026*, China;*
[2]*State key Lab of Fabrication Technologies for Integrated Circuits, Institute of Microelectronics, Chinese Academy of Sciences, Beijing* 100029*, China;*
[3]*Frontier Institute of Chip and System, State Key Laboratory of ASIC and System, Fudan University, Shanghai* 200433*, China;*
[4]*School of Microelectronics, University of Chinese Academy of Sciences, Beijing* 100049*, China*

Received 18 June 2022/Revised 2 September 2022/Accepted 17 November 2022/Published online 4 July 2023

Abstract Recent advances in developing beyond von Neumann architectures have moved the memristive devices to the forefront as one of the key enablers to realizing memristive computing-in-memory (mCIM) structures, which shows a great promise to boost the energy-efficiency and the performance of artificial intelligence (AI) chips. In this study, by considering the interactions between devices, circuits, and systems in the mCIM design, we propose several cross-layer design techniques, including (1) the BL-SL interactive forming protection (BSIFP) circuit that can reduce the voltage drop on the selected transistor, suppress the current overshoot by 65.96%, and improve the bit-cell density by more than 10.19%, (2) the clamping transistor trimming scheme (CTTS) to prevent the multiply-and-accumulate (MAC) signal margin degradation from chip-to-chip resistance variations, and (3) dynamic input-parallelism and output-precision (DIPOP) that can reduce the energy cost by 22.92% in a typical inference task with negligible accuracy loss. The results demonstrate the significant role of the cross-layer-interactive approach and provide a preliminary guideline for highly-efficient mCIM design.

Keywords memristor, resistive memory, computing-in-memory, artificial intelligence, cross-layer co-design

Citation An J J, Wang L F, Ye W, et al. Design memristor-based computing-in-memory for AI accelerators considering the interplay between devices, circuits, and system. Sci China Inf Sci, 2023, 66(8): 182404, https://doi.org/10.1007/s11432-022-3627-8

1 Introduction

In the relentless pursuit of high performance and low power computation, the energy and latency consumed by the data transferring between the processor and memory have been proven as a major bottleneck in von Neumann architectures [1]. The concerns on this issue have been further intensified by the rapid development of artificial intelligence (AI) technology [2,3]. Although it has excelled at a broad range of recognition and classification tasks associated with images, speeches, and objects [4], AI model sizes have been increasing exponentially over the years [5,6]. Consequently, the memory access and data movement required to process these models tend to substantially increase, which poses a critical challenge to designing efficient AI chips considering the von Neumann bottleneck.

Emerging nonvolatile memories (NVMs), such as resistive random-access memory (RRAM), phase change memory (PCM), and magnetic random-access memory (MRAM), have aroused extensive attention to boost the performance of next-generation AI chips [7–10]. Because of their non-volatility, high density, low power, and high speed, emerging NVMs are already being intensively studied as the embedded

* Corresponding author (email: douchunmeng@ime.ac.cn)

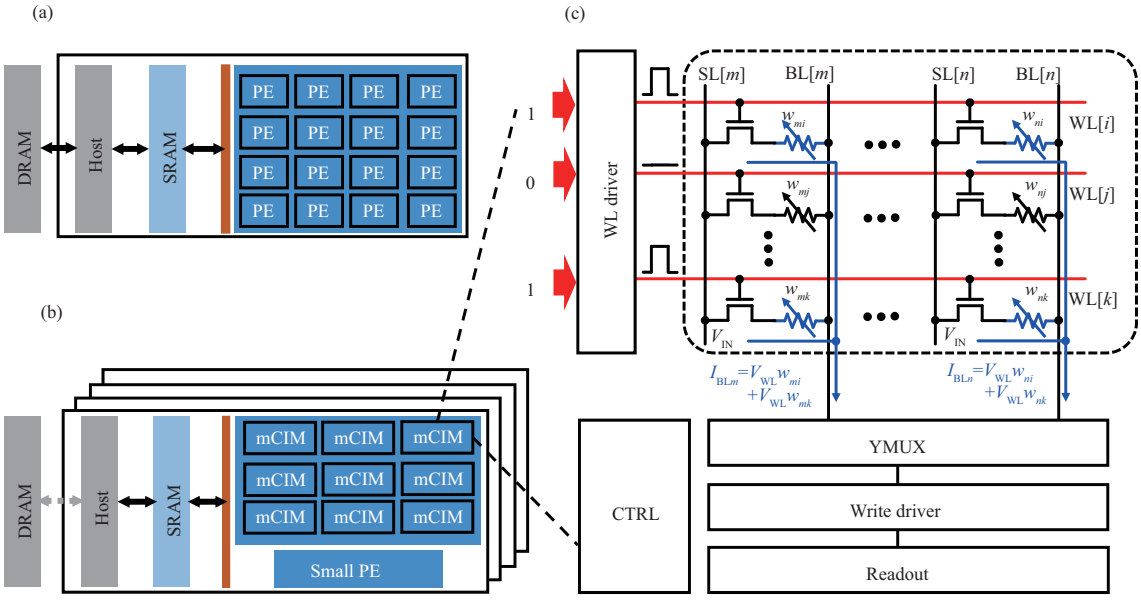
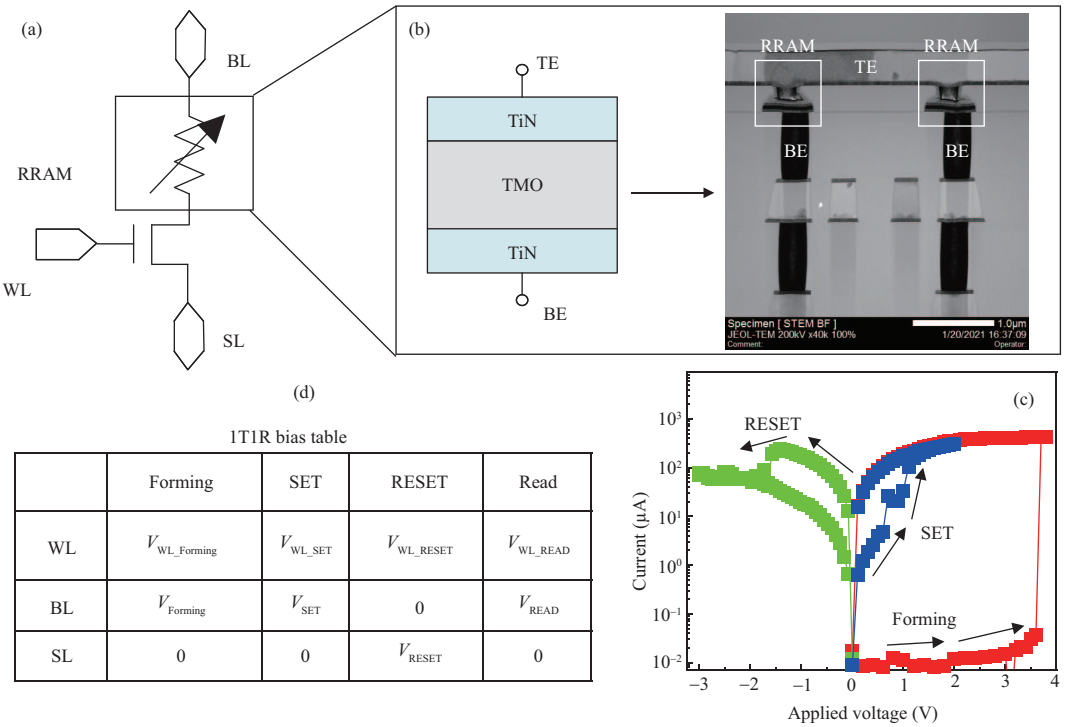


Figure 1 (Color online) Conceptual views of (a) von Neumann DNN processors, (b) mCIM-based DNN processors, and (c) mCIM macro structures.

NVMs in AI accelerators to achieve on-chip parameters accommodation [11, 12]. These advantages also make emerging NVMs become ideal technological platforms to build mCIM chips. Among different types of emerging NVMs, RRAM features good compatibility with advanced complementary metal oxide semiconductor (CMOS) technology, low extra integration cost, small latency, low operational voltage, and moderate resistance-ratio (R-ratio), which is attractive to developing embedded mCIM-based AI accelerators [13–24].

Figure 1(a) conceptually shows a typical deep neural network (DNN) processor based on the von Neumann architecture. It comprises a large array of processing elements (PEs) to execute arithmetic and logic operations in parallel, on-chip static random access memory (SRAM) to cache the input, output and weight data, off-chip dynamic random access memory (DRAM) to accommodate all weight data as well as the intermediate data, and the host as the control and interface of the system. Its performance is considerably bottlenecked by the memory accessing because (1) the weight data need to be fetched from the off-chip DRAM, and (2) large amounts of intermediate data generated in DNN processing need to be repeatedly written and read from the off-chip DRAM. As a result, the energy efficiency and the throughput of DNN processors tend to be limited by the energy and latency caused by cross-memory-hierarchy data movement.

On the other hand, the memristive computing-in-memory structure demonstrates great promise to bypass the von Neumann bottleneck [25–27]. Figure 1(b) conceptually shows the structure of mCIM-based DNN processors. It is composed of multiple mCIM macros as multiply-and-accumulate (MAC) engines, a small PE to process other algorithmic operations beyond MACs, the SRAM buffer and the host. The mCIM architecture can boost the efficiency of AI processors in several aspects. Firstly, mCIM macro can perform highly energy-efficient and paralleled analog MAC operations. Secondly, all of weight data can be possibly accommodated in the mCIM macros, hence the off-chip weight data fetching can be potentially eliminated. Thirdly, intermediate data that are produced during DNN processing can be effectively reduced. Particularly, the partial-sum results can be directly summed inside/near the mCIM macros. Figure 1(c) further shows the typical structure of mCIM macro, including one-transistor-one-resistor (1T1R) cell array, word-line (WL) driver, column multiplexer (YMUX), write driver, readout circuit and the control (CTRL) logics. In the computing mode, the activation data are input by applying voltage on WL (V_{WL}) and the weight data (w) are pre-stored in the cell array. Consequently, the MAC results can be readout from the data-line (DL), which can be either the bit-line (BL) or the source-line (SL) of the cell array. Notice that RRAM-based mCIMs can already fully accommodate many tiny AI models on-chip at present [28]. In order to process large AI models, large-scale mCIM systems composing multiple mCIM chips can be developed by using advanced integration technologies, such as 2.5D and 3D integration [29].



	Forming	SET	RESET	Read
WL	$V_{\text{WL\_Forming}}$	$V_{\text{WL\_SET}}$	$V_{\text{WL\_RESET}}$	$V_{\text{WL\_READ}}$
BL	V_{Forming}	V_{SET}	0	V_{READ}
SL	0	0	V_{RESET}	0

Figure 2 (Color online) (a) 1T1R bit-cell structure; (b) RRAM cell structure and cross-sectional views of the fabricated RRAM cells using the 180 nm CMOS process (TE and BE represent top and bottom electrodes); (c) typical bipolar resistive switching *I*-*V* curves, including the forming, set, and reset processes; (d) the biasing conditions for different operations.

Although the potential advantages of RRAM-based mCIMs have been demonstrated by many pioneer works, however, RRAM-based mCIMs are still facing several critical challenges, such as device non-idealities, small signal margins, limited parallelism, and readout precision. At the device level, the large voltage and overshoot current in the forming process hinder the scaling down of the accessing transistor [30]. At the circuit level, the signal margin (SM) in the MAC mode is much smaller than that in the memory mode, hence necessitating dedicated circuitry to optimize the SM of MAC signals [19]. At the system level, the widely existing sparsity in both the input and weight data can degrade the efficiency of the conventional full-precision readout schemes [24]. These challenges have necessitated a cross-layer design approach from device to system. In this study, we propose several interactive design techniques, including (1) BL-SL interactive forming protection (BSIFP) to improve bit-cell density and suppress over-shoot current, (2) CTTS to fine-tune MAC SM depending upon cell resistance, and (3) dynamic input-parallelism and output-precision (DIPOP) to boost the energy-efficiency as well as the throughput for DNN processing. The results demonstrate the significant role of the cross-layer-interactive approach as well as provide a preliminary guideline for highly-efficient mCIM design. The rest of this paper is organized as below. In Section 2, we present the proposed techniques by the co-designed device, circuit, and system. In Section 3, the performance of the proposed schemes is evaluated. Section 4 concludes this paper.

2 Proposed device-circuit-system interactive design techniques

2.1 Typical RRAM characteristics and BL-SL interactive forming protection scheme

Figure 2(a) shows the structure of a typical 1T1R RRAM bit-cell with transition-metal-oxide (TMO) based RRAM cell. The RRAM cell has a metal-insulator-metal sandwiched structure as shown in Figure 2(b), which can easily be integrated between the metal layers by the back-end-of-line (BEOL) process with high density. Figure 2(c) shows its typical resistive switching *I*-*V* curves. The forming process is first performed to activate the cell by applying $V_{\text{WL\_Forming}}$ and V_{Forming} to the word line (WL) and the bit line (BL), respectively. Then, the cell resistance can be reversibly switched by the reset and set process. The reset process can be done by applying $V_{\text{WL\_RESET}}$ and V_{RESET} to the WL and SL,

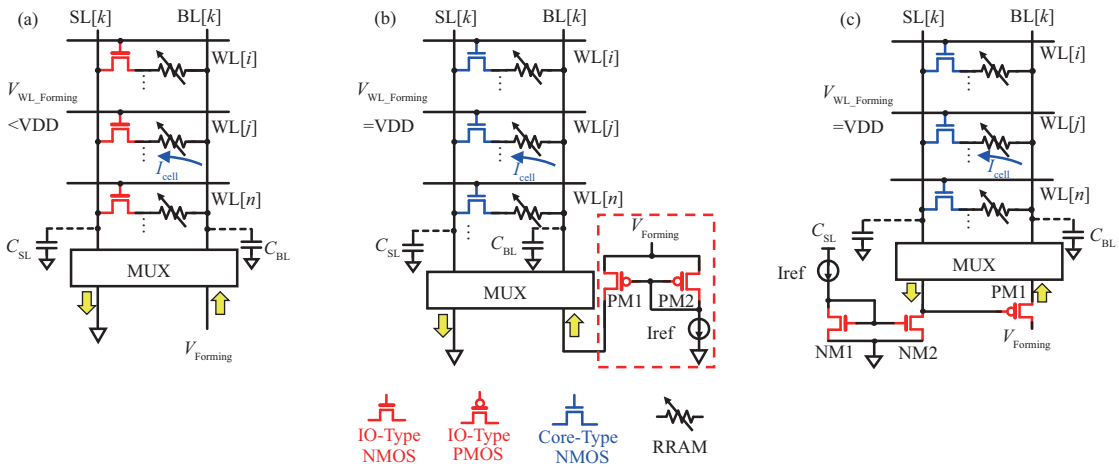


Figure 3 (Color online) Comparison between different forming schemes with different current compliance methods, including using (a) IO selector devices, (b) an additional IO current mirror, and (c) the proposed BL-SL interactive forming protection (BSIFP) scheme.

respectively. Similarly, the set process is carried out by applying $V_{\mathrm{WL\_SET}}$ and V_{SET} to the WL and BL, respectively. In the read process, the access transistor is activated by $V_{\mathrm{WL\_READ}}$ and V_{READ} is applied to the WL and BL, respectively. The biasing conditions for forming, set, reset, and read are summarized in Figure 2(d). It is noteworthy that V_{Forming} is usually much larger than V_{SET} or V_{RESET}. Hence, high voltage transistors for input and output (IO) are required in the 1T1R cells to tolerate the large V_{Forming}, which hinders the scaling down of the accessing transistor and limits the bit-cell density [31]. Besides, the large capacitive surge currents in the forming process may lead to a large overshoot current, which degrades cell reliability [32]. Therefore, it is necessary to introduce forming protection schemes to protect the bit-cells from large voltage drops and high overshoot currents during the forming process.

Figure 3 comparatively shows three types of forming schemes, including using the IO selectors [33], an additional current mirror [34, 35], and the proposed BSIFP schemes. In the IO selectors scheme (Figure 3(a)), the cell currents in the forming process are limited by controlling the gate voltage of the IO selectors. Although it can strictly limit the cell current, the selector devices suffer from a large voltage drop after switching the cells to the low resistive state (LRS), which rules out the possibility of using the area-efficient core transistors as the selectors. On the other hand, in the current mirror (CM) scheme (Figure 3(b)), the current compliance is achieved by controlling the reference current (I_{REF}) of the CM composed by the IO devices (PM1 and PM2), which reduce the voltage drop on the 1T1R bit-cell after the forming process. As a result, it is possible to use core devices as selectors. However, a remaining problem is because of the capacitive surging current between BL and SL, cell current overshoot can happen and incur reliability problems. To deal with the challenge, we proposed the BSIFP scheme (Figure 3(c)). In the forming operation, while the SL of the selected bit-cell is connected to an IO CM (NM1 and NM2) for current compliance, the BL of the cell is connected to V_{Forming} through an IO transistor (PM1) whose gate is connected to SL. Once the forming process is finished, the increasing SL voltage partially turns off the PM1, which helps to stabilize the cell currents and suppress the current overshoot.

2.2 Typical MAC behavior and clamping transistor trimming scheme

One of typical MAC circuits used in the voltage-mode mCIM macros [22, 24] is shown in Figure 4(a). The conventional voltage clamper uses a signal transistor with a fixed size. Depending upon the WL input data pattern and the stored weight data pattern, the MAC results can be readout from the data-line (DL) voltage. The input/weight data encoding methods are summarized in Figure 4(e). Figure 4(b) shows a typical computing cycle, and the SL is charged to 0.6 V. The BL is pre-charged to 0.2 V by V_{PRE}. The gate voltage of the clamping transistor (V_{CLP}) is set to 0.8 V. Then, the WLs are activated according to the input data patterns. Consequently, the DL voltage, corresponding to different MAC values (MACV), is determined by the voltage division between the parallel-connected RRAM cells with activated WLs and the clamping transistor. Then, the MACV results can be readout by the multi-level voltage-type sense amplifier (ML-VSA).

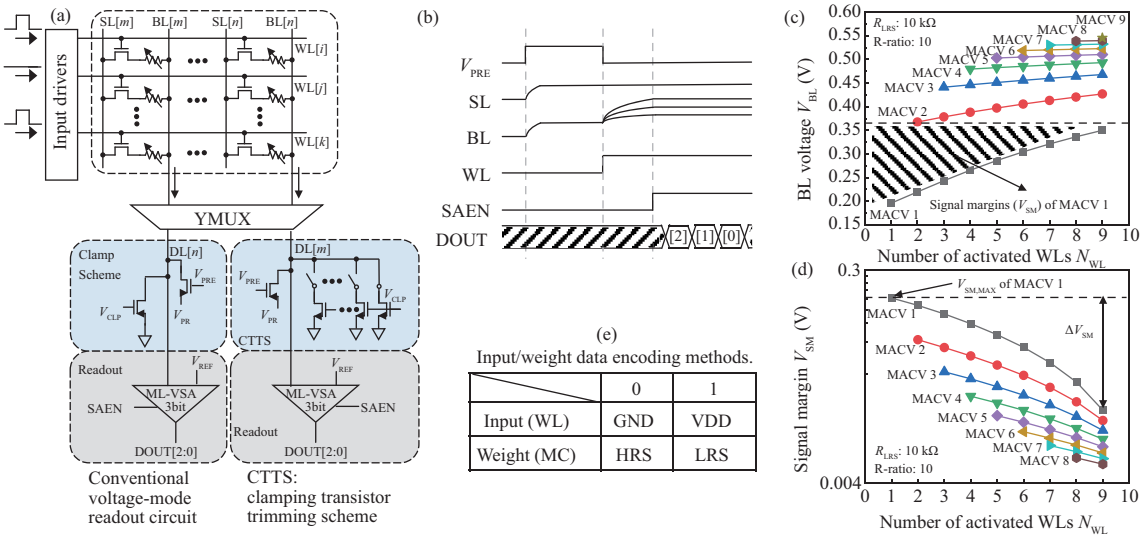


Figure 4 (Color online) (a) Typical mCIM macros with voltage-mode readout circuitry using conventional and proposed CTTS clamping schemes; (b) operation waveforms in a typical computing cycle; (c) BL voltage ($V_{\rm BL}$) distribution as a function of different number of activated WLs ($N_{\rm WL}$) corresponding to different MACVs; (d) the read signal margin ($V_{\rm SM}$) of different MACVs; (e) the input and weight data encoding methods.

The typical distribution of the MAC signals, in the case of 9 parallel WL inputs, is evaluated based on typical foundry RRAM properties. The BL voltage corresponding to different MACVs as a function of the number of activated WLs ($N_{\rm WL}$) is shown in Figure 4(c). The read signal margin ($V_{\rm SM}$) of a given MACV can be defined as its voltage difference from the neighboring one. The $V_{\rm SM}$ is also dependent on the data pattern (Figure 4(d)). The maximum $V_{\rm SM}$ ($V_{\rm SM,MAX}$) of an MACV can be observed when no spare high resistance state (HRS) cells are activated and it decreases with increasing activated HRS cells. The results indicate that to differentiate the MAC signals and improve input parallelism, it is of particular importance to optimize $V_{\rm SM,MAX}$. In order to achieve optimal $V_{\rm SM,MAX}$, we propose the clamping transistor trimming scheme (CTTS), in which multiple clamping transistors are used in the voltage clamper. By adjusting the number of clamping transistors ($N_{\rm CLP}$), it is capable to fine-tune $V_{\rm SM,MAX}$ according to the cell resistances. Further details will be discussed in Section 3.

2.3 The weight mapping and dynamic input parallelism and output precision scheme

Figure 5(a) shows a typical DNN model, consisting of multi-layer convolution (CNN) and fully connected neural network (FCNN). Here, while the CNN layer can extract feature maps of the input data through a high-level abstraction, the FCNN layer can classify the features into different categories. In a CNN layer, a convolution operation between the input and the kernel (or filter) weights is performed first, and then the output is given after pooling and activation. The mapping method of the weight data of the convolutional layer is shown in Figure 5(b). The weight data in a $k \times k$ convolution kernel are unrolled and mapped to k^2 cells in the same column. Notice that mapping different kernels corresponding to the same output channel into the same mCIM macro can reduce the data movement between different macros. On the other hand, in an FCNN layer, the output is given by the weighted sum of the input data after activation. Figure 5(c) shows the mapping method of the weights of the FCNN layer. The weight data corresponding to different input neurons and the same output neuron are mapped into the same column. If the number of input neurons is larger than the number of rows of the cell array, the weight data can be mapped to different columns and selected by the column multiplexer (MUX) in different clock (CLK) cycles.

Based on the above weight data mapping scheme, we have further proposed the dynamic input parallelism and output precision (DIPOP) DNN accelerating scheme. In a compute-in-memory (CIM) macro with binary weight and input, the relationship between the input parallelism ($N_{\rm IN}$) and read readout precision (b_O) can be given by $b_O = \log_2(N_{\rm IN} + 1)$. Conventionally, fixed $N_{\rm IN}$ and b_O are used for both CNN and FCNN processing (Figure 6(a)). However, because of the commonly existing sparsity in both input and weight data, the number of practically appeared MACVs is usually much smaller than $\log_2(N_{\rm IN} + 1)$, which degrades the utilizing efficiency of the macros. In the proposed DIPOP scheme, we

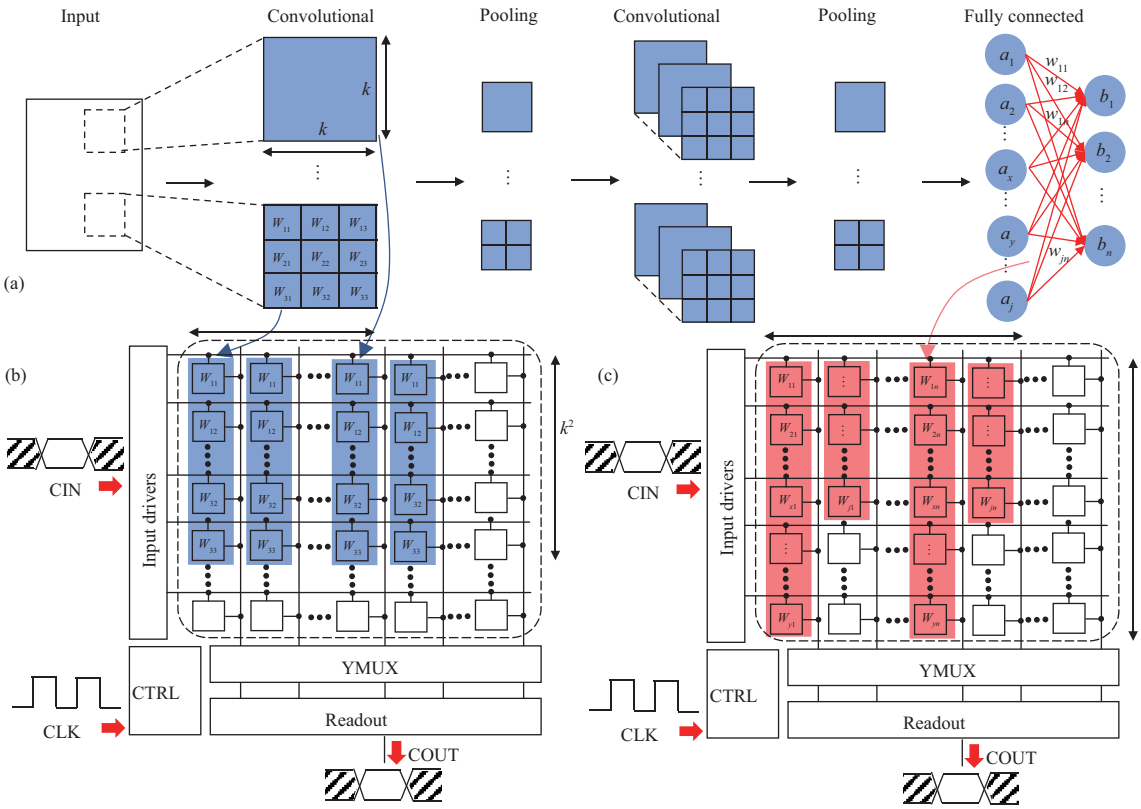


Figure 5 (Color online) (a) A typical DNN model structure, (b) mCIM weight mapping for CNN layers, and (c) mCIM weight mapping for FCNN layers. k, CIN, and COUT refer to the size of the kernel, the input, output channels/neurons of the CNN/FCNN connected layers.

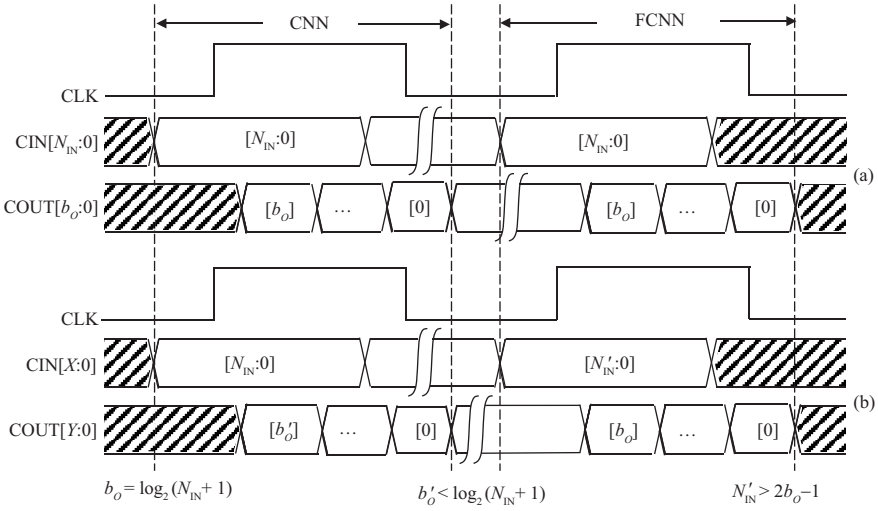


Figure 6 Conceptual illustrations of the input parallelism and output precisions for CNN and FCNN in (a) the conventional dataflow and (b) the proposed dynamic input parallelism and output precision (DIPOP) scheme.

dynamically set the readout precision and input parallelism for CNN and FCNN processing (Figure 6(b)). In a typical computing cycle for CNN processing, N_{IN} equals $k \times k$ for kernel-order computing, and we use a reduced b_O (b'_O), which is smaller than $\log_2(N_{\mathrm{IN}}+1)$, to avoid redundant analog quantization process for energy saving. In a typical cycle for FCNN, we use a boosted N_{IN} (N'_{IN}), which is larger than $(2^{b_O}-1)$, to fully utilize the implemented readout circuit. By leveraging the DIPOP scheme, the CIM macros can process DNN with reduced energy cost and improved throughput with a negligible accuracy loss.

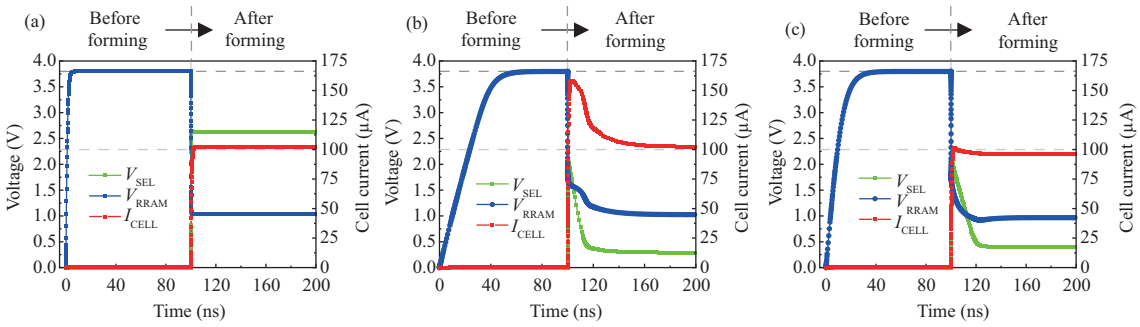


Figure 7 (Color online) Comparisons on the voltage drop on the selectors (V_{SEL}), the voltage drops on the RRAM cell (V_{RRAM}), and the current passed through the RRAM cell (I_{CELL}) before and after the forming process in different forming schemes, including (a) the IO selector devices, (b) an additional IO current mirror, and (c) the proposed BSIFP scheme.

3 Results and discussion

3.1 The impact of the BSIFP scheme

In Figure 7, we compare the voltage drop on the RRAM cells (V_{RRAM}) as well as the selected transistors (V_{SEL}) and the cell currents in the forming process using different forming protection and current compliance schemes, including (a) the IO selectors, (b) the CM, and (c) the proposed BSIFP circuits. We use the properties of typical TMO RRAM cells for the simulations, which have a cell resistance (R_{CELL}) of 10 MΩ at the fresh state and 10 kΩ at LRS and can be formed by applying 3.8 V for 100 ns. The array size is 1024 × 1024 with a parasitic capacitance of 1 pF on each BL and SL. The forming process is carried out with a target compliance current of 100 μA.

In the IO selector forming scheme (Figure 7(a)), the V_{WL} of the selector is set to 1.35 V to limit the cell current. It can be seen that although the IO selector device can strictly limit the cell current, it incurs large V_{SEL} after forming and results in difficulties to scale down the IO transistor. Figure 7(b) shows the case of using an additional IO CM made by IO devices in the write drivers for forming, which limit the current by applying a reference current (I_{REF}) of 100 μA. After the forming process, V_{RRAM} and V_{SEL} can be effectively reduced because there is a considerable voltage drop on the IO device in the CM. The final V_{SEL} can be lower than 0.5 V, which makes using core logic devices as the selectors become possible. However, it suffers from an overshoot current of up to 158.44 μA at the moment of resistive switching and requires a relatively long time about 67 ns for stabilizing the cell current because of the capacitive surge current induced by the BL (C_{BL}) and SL capacitance (C_{SL}). The large overshoot current can result in the irreversible hard breakdown of the cell [32]. In the proposed BSIFP circuit (Figure 7(c)), the current compliance is also realized by applying an I_{REF} of 100 μA. Before the forming process is finished, the SL voltage is low and PM1 is fully activated to pass the V_{Forming} without lowering V_{RRAM}. After switching to the LRS, the SL voltage rise and hence partially turn off PM1, which helps to stabilize the cell current. Because of the BSIFP scheme, cell current overshoot can be avoided while maintaining low V_{SEL} after forming.

By lowering the V_{RRAM} and suppressing the overshoot current in the forming process, the BSIFP circuit makes it possible to use the core logic device as the selector. Figure 8(a) compares the layout of 180 nm 400 Kb mCIM macros using IO selectors and core selectors with BSIFP. It shows that by using the BSIFP scheme, the area of cell array can be reduced by 13.54% and that of the whole macro can be reduced by 10.19%. Furthermore, considering the difference in area between IO and core devices continually increases as technology node evolves, the impact of the BSIFP scheme tends to increase at the advanced technology nodes. We have further evaluated its impact based on the foundry process develop kits (PDKs), as shown in Figure 8(b). The BSIFP can potentially lead to a 51.53% reduction of mCIM macro sizes at the 14 nm technology nodes.

3.2 Analysis of MAC signal margins and the CTTS

As previously discussed, optimizing the MAC signal margin is of particular importance for mCIM. Considering the circuit shown in Figure 4(a) and the voltage division relationship between the RRAM cells and the clamping transistor. The maximum MAC signal margin ($V_{\mathrm{SM,MAX}}$) of a given MACV can be

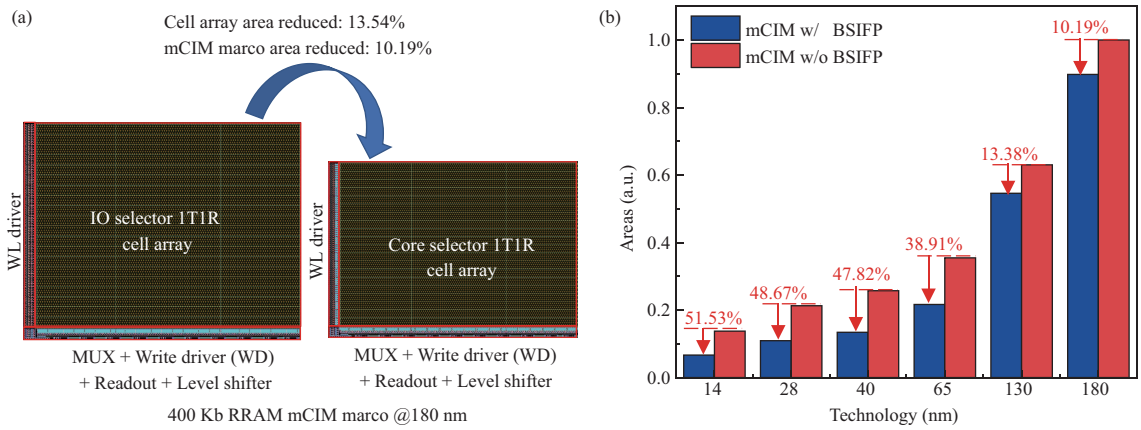


Figure 8 (Color online) (a) Comparisons on the areas between 180 nm 400 kb mCIM macros with and without the BSIFP scheme, and (b) evaluation of the impact of BSIFP at the advanced technology nodes. The BSIFP can improve the area efficiency up to 51.53% by replacing the IO selectors with the core selectors.

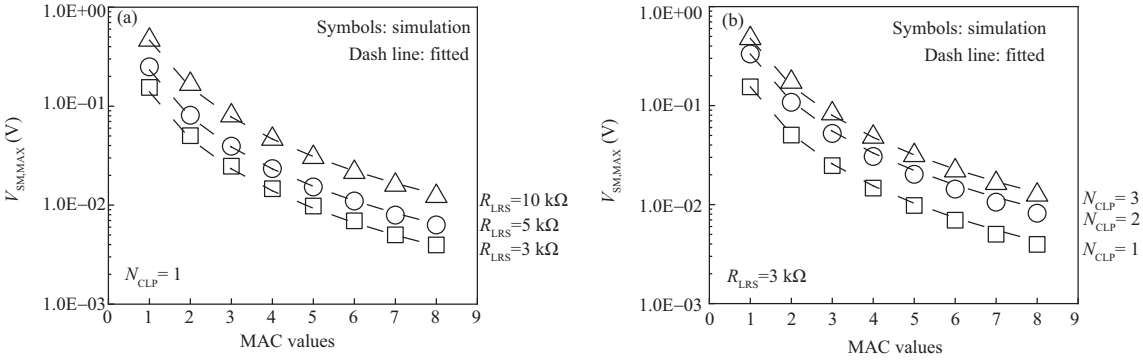


Figure 9 (Color online) Maximum V_{SM} as a function of MACVs with (a) different LRS resistances (R_{LRS}) and (b) different numbers of the clamping transistors (N_{CLP}). The symbols show the SPICE simulation results and the dash lines show the fitted results given by the empirical equations.

given by

$$V_{SM,MAX} = \alpha \frac{\Delta MACV}{MACV(\Delta MACV + MACV)} R_{LRS} N_{CLP}, \quad (1)$$

where α is the fitting parameter, ΔMACV is the space between the neighboring MACV, and N_{CLP} is the number of fingers of the clapping transistor. It indicates that the MAC signal margin can be optimized by increasing either the cell resistance or the width of the clamping transistor. The derived relationship is verified by fitting (1) with the SPICE simulated signal margins, as shown in Figure 9. It can be seen that $V_{SM,MAX}$ nearly quadratically decreases with increasing MACV and are proportional to R_{LRS} and N_{CLP}. Considering the differences between cell resistance in mass production, the average R_{LRS} can be different from chip to chip. By trimming the strength of the clamping transistor at the macro level, the CTTS can effectively optimize the signal margins.

3.3 Performance evaluations of the DIPOP scheme

To evaluate the performance of DIPOP at the system level, we perform a case study using a binary LENET model for MNIST digits classification. Figure 10(a) shows the distribution of MAC values for CNN processing using mCIM macros. Here, we choose $N_{IN} = 9$ to process the 3×3 kernel in a signal mCIM cycle. It can be seen that because of the sparsity in both the input and weight data, most of MACVs are less than 4 (97.3% in C1 and 99.3% in C2). Although the full readout precision is 4-bit, a readout precision of 2-bit can be sufficient to differentiate most of MACVs in practice. Figure 10(b) further shows the distribution of MACs for FCNN layers processing using mCIM macros with N_{IN} ranging from 9 to 90. It shows that even in the case of $N_{IN} = 36$, most of MACVs are less than 8 (99.9% in FC1 and 96.6% in FC2), which indicates that an output precision of 3-bit would be sufficient.

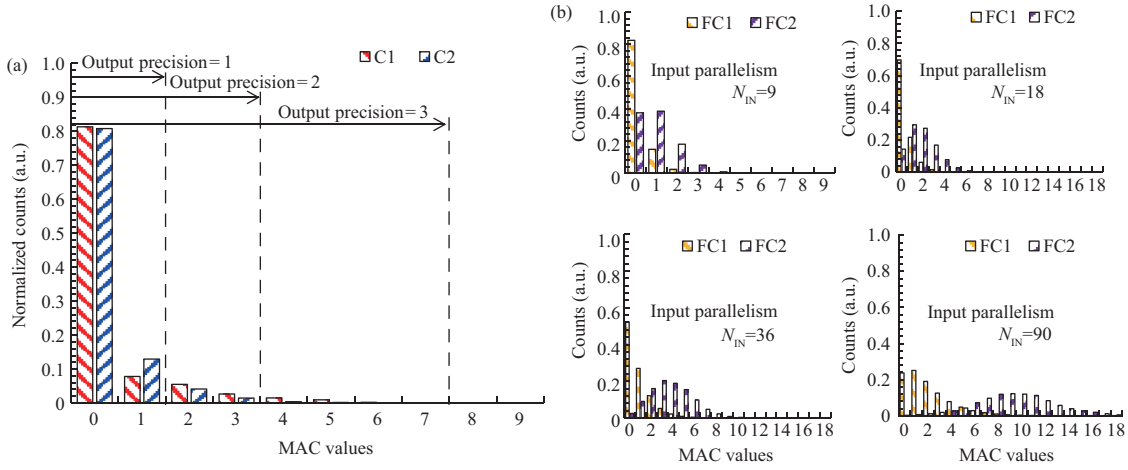


Figure 10 (Color online) (a) Distribution of MAC values for CNN layers processing with $N_{IN} = 9$. Here, C1 and C2 refer to the first and second CNN layers, respectively. (b) Distribution of MAC values for FCNN layers processing with $N_{IN} = 9$, 18, 36, and 90. Here, FC1 and FC2 refer to the first and second FCNN layers, respectively.

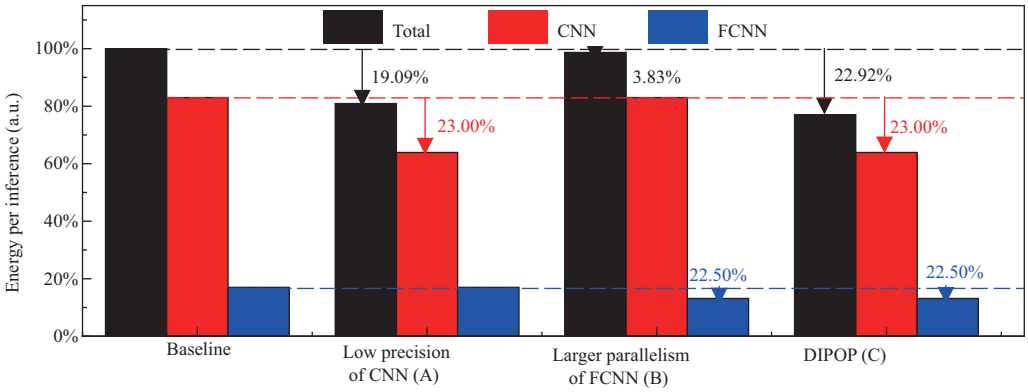


Figure 11 (Color online) Energy per inference task based on different strategies. The DIPOP scheme can reduce the average energy per inference task by 22.92% in total.

Table 1 Comparison between different DNN processing schemes using mCIMs with different input parallelism and output precision

	Baseline		Lower precision of CNN (A)		Larger parallelism of FCNN (B)		DIPOP (C)	
	Input parallelism	Output precision	Input parallelism	Output precision	Input parallelism	Output precision	Input parallelism	Output precision
C1	9	3	9	2	9	3	9	2
C2	9	3	9	2	9	3	9	2
FC1	9	3	9	3	36	3	36	3
FC2	9	3	9	3	36	3	36	3
Inference accuracy (%)	98.82		98.78		98.65		98.63	

Different implantations of input parallelism and output precisions are summarized in Table 1. Compared to the baseline conditions with full precision readout, reducing the output precision to 2-bit for CNN processing and increasing the input parallelism to 36 for FCNN processing result in a negligible accuracy loss of 0.19%. The power consumption and throughput of the mCIMs can considerably benefit from reducing the output precision and increasing the input parallelism. Particularly, reducing the output precision can effectively reduce the power and latency consumed for the analog-to-digital convention (ADC) process. Besides, improving the input parallelism can reduce the number of computing cycles required for an inference task. It not only reduces the runtime for the task, but also reduces the total number of analog readouts. Figure 11 compares the energy required for a single inference under different strategies. It can be seen that by using the DIPOP scheme, the energy for CNN and FCNN processing can be reduced by 23.00% and 22.50%, respectively, and the energy for an inference task can be reduced by 22.92%. Besides, by improving N_{IN} from 9 to 36 in FCNN layers, the throughput for the FCNN

process can be improved by 4 times.

4 Conclusion

In this study, we have provided a cross-layer perspective on mCIM design from devices, and circuits to the system. On this basis, we first proposed the BSIFP scheme, which can reduce the voltage drop on the selector devices and suppress the overshoot current by 65.96% in the forming process. A case study on a 180 nm 400 kb CIM macro design shows the BSIFP can shrink the macro size by 10.19%. Then, we propose the CTTS to prevent the MAC signal margin degradation from chip-to-chip resistance variations in mass production. Lastly, we proposed the DIPOP scheme to boost the performance of mCIM macro leveraging the sparsity in DNN models. A case study shows it can reduce the energy cost by 22.92% in a typical inference task with negligible accuracy loss. The results reveal that the cross-layer design methodology can play an important role in future design and optimizing the performance, accuracy, and energy efficiency of mCIM-based AI accelerators.

Acknowledgements This work was supported by National Key R&D Program of China (Grant Nos. 2018YFA0701500, 2018YFB-2202900), National Natural Science Foundation of China (Grant Nos. 61904197, 61934005, 61825404, 61732020, 61821091, 61888102), Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB44000000), and Project of MOE Innovation Platform.

References

1 Patterson D. 50 years of computer architecture: from the mainframe CPU to the domain-specific TPU and the open RISC-V instruction set. In: Proceedings of IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, 2018. 27–31
2 Sze V. Designing hardware for machine learning: the important role played by circuit designers. IEEE Solid-State Circuits Mag, 2017, 9: 46–54
3 Xu X, Ding Y, Hu S X, et al. Scaling for edge inference of deep neural networks. Nat Electron, 2018, 1: 216–222
4 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444
5 Amodei D, Hernandez D, Sastry G, et al. AI and compute. 2019. https://openai.com/blog/ai-and-compute/
6 Dou C, Xu X, Zhang X, et al. Enabling RRAM-based brain-inspired computation by co-design of device, circuit, and system. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), San Francisco, 2021. 21–24
7 Yu S, Chen P-Y. Emerging memory technologies: recent trends and prospects. IEEE Solid-State Circuits Mag, 2016, 8: 43–56
8 Zidan M A, Strachan J P, Lu W D. The future of electronics based on memristive systems. Nat Electron, 2018, 1: 22–29
9 Ielmini D, Wong H S P. In-memory computing with resistive switching devices. Nat Electron, 2018, 1: 333–343
10 Dou C-M, Chen W-H, Xue C-X, et al. Nonvolatile circuits-devices interaction for memory, logic and artificial intelligence. In: Proceedings of IEEE Symposium on VLSI Technology, Honolulu, 2018. 171–172
11 Wan W, Kubendran R, Eryilmaz S B, et al. A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models. In: Proceedings of IEEE International Solid- State Circuits Conference (ISSCC), San Francisco, 2020. 498–500
12 Li Z, Wang Z, Xu L, et al. RRAM-DNN: an RRAM and model-compression empowered all-weights-on-chip DNN accelerator. IEEE J Solid-State Circuits, 2021, 56: 1105–1115
13 Su F, Chen W-H, Xia L, et al. A 462GOPs/J RRAM-based nonvolatile intelligent processor for energy harvesting IoE system featuring nonvolatile logics and processing-in-memory. In: Proceedings of Symposium on VLSI Technology, Kyoto, 2017. 260–261
14 Mochida R, Kouno K, Hayata Y, et al. A 4M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture. In: Proceedings of IEEE Symposium on VLSI Technology, Honolulu, 2018. 175–176
15 Jiang Y, Huang P, Zhu D, et al. Design and hardware implementation of neuromorphic systems with RRAM synapses and threshold-controlled neurons for pattern recognition. IEEE Trans Circuits Syst I, 2018, 65: 2726–2738
16 Cai F, Correll J M, Lee S H, et al. A fully integrated reprogrammable memristor-CMOS system for efficient multiply-accumulate operations. Nat Electron, 2019, 2: 290–299
17 Chen W H, Dou C, Li K X, et al. CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors. Nat Electron, 2019, 2: 420–428
18 Liu Q, Gao B, Yao P, et al. A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing. In: Proceedings of IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, 2020. 500–502
19 Xue C-X, Hung J-M, Kao H-Y, et al. A 22 nm 4 Mb 8b-precision ReRAM computing-in-memory macro with 11.91 to 195.7 TOPS/W for tiny AI edge devices. In: Proceedings of IEEE International Solid- State Circuits Conference (ISSCC), San Francisco, 2021. 245–247
20 Zhou K, Zhao C, Fang J, 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, 2021, 68: 2932–2936
21 Song T, Chen X, Zhang X, et al. BRAHMS: beyond conventional RRAM-based neural network accelerators using hybrid analog memory system. In: Proceedings of the 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, 2021. 1033–1038
22 Yoon J H, Chang M, Khwa W S, et al. A 40-nm, 64-Kb, 56.67 TOPS/W voltage-sensing computing-in-memory/digital RRAM macro supporting iterative write with verification and online read-disturb detection. IEEE J Solid-State Circuits, 2022, 57: 68–79
23 Hung J-M, Huang Y-H, Huang S-P, 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: Proceedings of IEEE International Solid- State Circuits Conference (ISSCC), San Francisco, 2022. 1–3

24 Li W, Sun X, Huang S, et al. A 40-nm MLC-RRAM compute-in-memory macro with sparsity control, on-chip write-verify, and temperature-independent ADC references. IEEE J Solid-State Circuits, 2022, 57: 2868–2877
25 Zhang W, Gao B, Tang J, et al. Neuro-inspired computing chips. Nat Electron, 2020, 3: 371–382
26 Zou X Q, Xu S, Chen X M, et al. Breaking the von Neumann bottleneck: architecture-level processing-in-memory technology. Sci China Inf Sci, 2021, 64: 160404
27 Yu S, Jiang H, Huang S, et al. Compute-in-memory chips for deep learning: recent trends and prospects. IEEE Circuits Syst Mag, 2021, 21: 31–56
28 Wan W, Kubendran R, Schaefer C, et al. A compute-in-memory chip based on resistive random-access memory. Nature, 2022, 608: 504–512
29 Zhu H, Jiao B, Zhang J, et al. COMB-MCM: computing-on-memory-boundary NN processor with bipolar bitwise sparsity optimization for scalable multi-chiplet-module edge machine learning. In: Proceedings of IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, 2022. 1–3
30 Portal J M, Bocquet M, Onkaraiah S, et al. Design and simulation of a 128 kb embedded nonvolatile memory based on a hybrid RRAM (HfO_2)/28 nm FDSOI CMOS technology. IEEE Trans Nanotechnol, 2017, 16: 677–686
31 Beckmann K, Holt J, Manem H, et al. Nanoscale hafnium oxide RRAM devices exhibit pulse dependent behavior and multi-level resistance capability. MRS Adv, 2016, 1: 3355–3360
32 Sekar D C, Bateman B, Raghuram U, et al. Technology and circuit optimization of resistive RAM for low-power, reproducible operation. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), San Francisco, 2014. 21–24
33 Wan W, Kubendran R, Schaefer C, et al. Edge AI without compromise: efficient, versatile and accurate neurocomputing in resistive random-access memory. 2021. ArXiv:2108.07879
34 Xue X Y, Jian W X, Yang J G, et al. A 0.13 μm 8 Mb logic based CuxSiyO resistive memory with self-adaptive yield enhancement and operation power reduction. In: Proceedings of IEEE Symposium on VLSI Circuits, Honolulu, 2012. 42–43
35 Jain P, Arslan U, Sekhar M, et al. A 3.6 Mb 10.1 Mb/mm^2 embedded non-volatile ReRAM macro in 22 nm FinFET technology with adaptive forming/set/reset schemes yielding down to 0.5 V with sensing time of 5 ns at 0.7 V. In: Proceedings of IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, 2019. 212–214