SCIENCE CHINA
Information Sciences

October 2023, Vol. 66 200408:1–200408:10
https://doi.org/10.1007/s11432-023-3785-8

• PROGRESS •
Special Topic: Recent Progress of Fundamental Research on Post-Moore Novel Devices

Architecture-circuit-technology co-optimization for resistive random access memory-based computation-in-memory chips

Yuyi LIU, Bin GAO*, Jianshi TANG, Huaqiang WU & He QIAN

School of Integrated Circuits, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084*, China*

Received 30 January 2023/Revised 31 March 2023/Accepted 26 May 2023/Published online 18 September 2023

Abstract Computation-in-memory (CIM) chips offer an energy-efficient approach to artificial intelligence computing workloads. Resistive random-access memory (RRAM)-based CIM chips have proven to be a promising solution for overcoming the von Neumann bottleneck. In this paper, we review our recent studies on the architecture-circuit-technology co-optimization of scalable CIM chips and related hardware demonstrations. To further minimize data movements between memory and computing units, architecture optimization methods have been introduced. Then, we propose a device-architecture-algorithm co-design simulator to provide guidelines for designing CIM systems. A physics-based compact RRAM model and an array-level analog computing model were embedded in the simulator. In addition, a CIM compiler was proposed to optimize the on-chip dataflow. Finally, research perspectives are proposed for future development.

Keywords resistive random-access memory, computation-in-memory, compact model, device-architecture-algorithm co-design, compiler

Citation Liu Y Y, Gao B, Tang J S, et al. Architecture-circuit-technology co-optimization for resistive random access memory-based computation-in-memory chips. Sci China Inf Sci, 2023, 66(10): 200408, https://doi.org/10.1007/s11432-023-3785-8

1 Introduction

Many artificial intelligence (AI) applications, such as speech recognition, computer vision, and recommendation systems, have achieved significant progress. This is primarily attributed to the development of deep learning algorithms. These tremendous developments have also prompted innovation in AI chip architectures. Currently, customized hardware architectures are available for AI, particularly for deep learning, to improve hardware performance and energy efficiency. CPUs, GPUs, and ASICs are optimized for AI applications to achieve better real-time response speeds and low power consumption. However, these CMOS-based chip architectures have inevitable data exchange between memory and computing units, and memory access is much more expensive than computing power consumption [1, 2]. This is known as the "memory wall" of the von Neumann system [3]. Because deep neural networks have a large number of parameters, CMOS-based accelerators have obvious bottlenecks in improving performance and energy efficiency.

The resistive random-access memory (RRAM, also called memristor)-based computation-in-memory (CIM) architecture can integrate the memory unit and the computing unit as the basic processing unit [4–6]. This is analogous to biological brains in that neurons and synapses in biological brains can store and process information in parallel while processing cognitive tasks with low power. RRAM is a type of nonvolatile memory (NVM) with high storage density, low read power consumption, and good analog programmability [2, 4, 5, 7, 8]. RRAM arrays can accelerate vector-matrix-multiplication (VMM) operations in neural network computations with high energy efficiency. However, frequent data transfers between RRAM arrays and off-chip memory result in significant latency. In addition, the area and

* Corresponding author (email: gaob1@tsinghua.edu.cn)

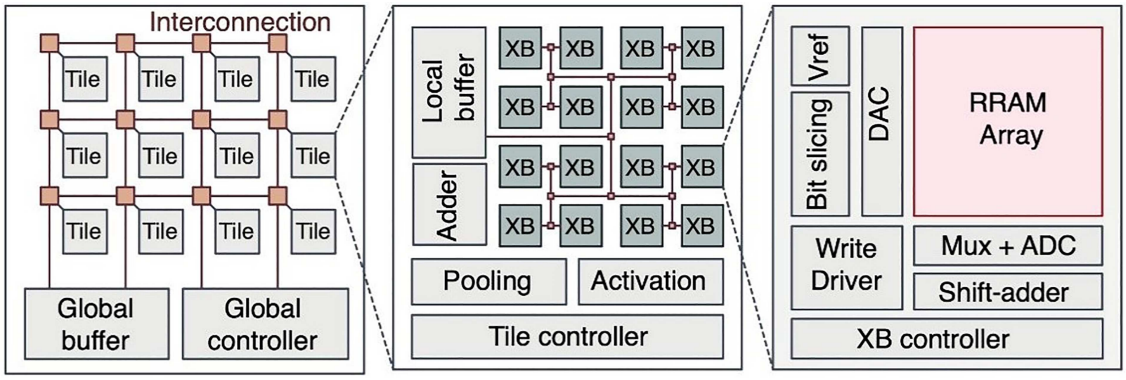


Figure 1 Architecture of the scalable CIM chip.

power overhead of digital-to-analog converters (DACs)/analog-to-digital converters (ADCs) also reduce system-level energy efficiency. The RRAM-based CIM architecture and demonstrations on systems are introduced in Subsections 2.1 and 2.2. The cross-level co-optimization of RRAM-based CIM systems is introduced in Subsection 2.3. In Subsection 2.4, we propose a monolithic-three-dimensional integration (M3D)-based hybrid CIM architecture to improve system-level energy efficiency and throughput.

Furthermore, RRAM-based CIM systems encounter certain challenges for neural network inference and online training. First, actual RRAM has nonideal characteristics and reliability issues [2,9] such as weight-update nonlinearity and asymmetry, limited endurance, and limited retention. These carry errors into the weight stored in the RRAM. Second, the IR drop caused by parasitic resistance and the offset and noise of peripheral circuits can also cause calculation errors [10]. Third, the trade-off between energy efficiency, flexibility, and computational accuracy cannot be addressed in a single abstraction layer. It is necessary to build a device-architecture-algorithm co-optimization simulator. Several simulator platforms for co-design have been proposed, such as NeuroSim [11, 12] and MNSIM [13, 14], which can perform data flow or circuit-level simulations. A physics-based RRAM compact model and an array-level analog computing model have been introduced in Subsection 3.1. We propose an RRAM-based hardware simulator to provide guidelines for designing CIM systems in Subsection 3.1 and a CIM compiler to optimize the on-chip dataflow in Subsection 3.2.

2 RRAM-based CIM architecture

2.1 Architecture of scalable CIM chip

To adapt to different scenarios and performance requirements, a CIM chip architecture with scalable computing performance is proposed. Each basic computing module has unit computing and storage resources, and the proportional growth of hardware resources can be achieved when mapping neural networks of different scales. The RRAM-based CIM architecture consists of three levels, namely, chip, tile, and basic processing unit (PE, also called crossbar or XB) [2, 15], as shown in Figure 1. The top level, or chip level, comprises multiple interconnected tiles and global units. Data between tiles are communicated through an on-chip interconnection fabric. The data flow is data-driven. Each tile initiates its calculation if sufficient data are sent to the local buffer of the tile. The second level, the tile-level, is for the convenience of hierarchical management. The tile-level is area-efficient because XBs in one tile can share some circuit units, such as adder-trees, pooling units, and activation units. A tile consists of several XBs, local buffers, a tile-level controller, and special function units (SFUs). The third level is the XB-level, which is composed of RRAM arrays and other peripheral circuit units, such as DACs, ADCs, and write drivers. For RRAM arrays, the weights of neural networks are represented as the conductance of RRAM cells, and the input feature maps (IFMs) are mapped to the voltage-level-based inputs encoded by DACs. The VMM can be implemented in RRAM arrays following Ohm's law and Kirchhoff's current law. The output currents of the columns are quantified by ADCs and then processed by SFUs. CIM chips can be scaled at the tile-level and XB-level to implement large-scale neural networks.

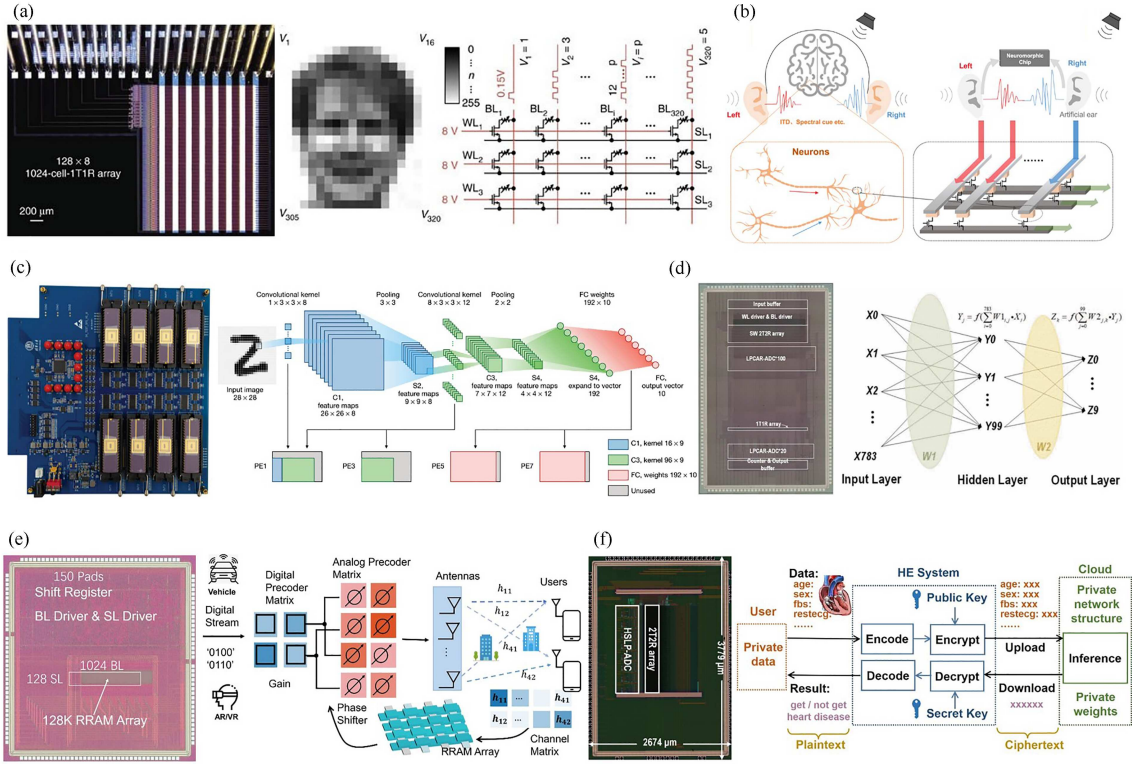


Figure 2 Various demonstrations are experimentally implemented on hardware systems. (a) The grayscale face image classification [16] and (b) sound localization on a 1k-1T1R array [17]; (c) a five-layer RRAM-based CNN on a fully hardware-implemented PCB-level multi-array integrated system [5] Copyright 2020 Nature; (d) a complete multi-layer FCNN on a 160 kb fully-integrated analog RRAM-based chip [19] Copyright 2020 IEEE; (e) a hybrid precoding technology for 5G/6G MIMO communication system on a fully parallel 128 Kb RRAM array [21] Copyright 2022 IEEE; (f) an encryption-decryption process for privacy on eight 144 kb 2T2R RRAM arrays [22] Copyright 2022 IEEE.

2.2 Demonstrations on RRAM-based CIM systems

To demonstrate the feasibility and efficiency of CIM and explore hardware operation modes under different application scenarios, various demonstrations are experimentally implemented on RRAM-based hardware systems. A fabricated 1k-1T1R array is trained online for grayscale face image classification [16]. Two online programming methods, a write-verify method and a without-write-verify method, are used for weight updates. This experimental demonstration has equivalent accuracy on test sets compared with a CPU. This consolidates the feasibility of RRAM arrays as analog synapses (Figure 2(a)). In addition, sound localization, a basic cognitive function of human beings, is also demonstrated on the 1k-1T1R array [17]. A multi-threshold-update scheme is proposed to make the in-situ training weights more stable and precise (Figure 2(b)). Then, a five-layer RRAM-based CNN (convolutional neural network) to perform MNIST image recognition with 96.19% accuracy is experimentally implemented on a fully hardware-implemented printed circuit board (PCB)-level multi-array integrated system [5]. This study proposes a hybrid training method. It is a system-level solution for increasing immunity to device imperfections. The method only needs to fine-tune the last fully connected (FC) layer of the neural network to achieve a tradeoff between the overall system performance and the overhead of learning energy. Replication of multiple convolutional (CONV) kernels to RRAM arrays is implemented to balance the processing speed between the CONV and FC layers. The benchmark shows more than two orders of magnitude better energy efficiency than the V100 GPUs (Figure 2(c)). In addition, binary morphology operations are demonstrated in the CIM integrated system [18]. The basic operations include dilation, erosion, opening, and closing. These morphological operations can be used for defect detection and medical image processing.

A 160 kb fully integrated analog RRAM-based chip for a complete multi-layer FCNN (fully connected neural network) is presented in [19], with sign-weighted 2T2R arrays to reduce IR-drop and power consumption. The chip realizes 94.4% accuracy on the MNIST dataset and 78.4 TOPS/W peak energy efficiency (Figure 2(d)). A four-layer Bayesian neural network is demonstrated on a 160 kb RRAM ar-

ray with 97% accuracy for MNIST image classification [20]. The distribution of the total current of N RRAM cells is represented by a weighted probability distribution in the Bayesian neural network. The RRAM-based network can also detect adversarial images using the RRAM intrinsic read noise.

A hybrid precoding technology for a 5G/6G MIMO communication system is demonstrated on a fully parallel 128 Kb RRAM array [21]. It can achieve a sum rate comparable to that of an FPGA (field programmable gate array) and higher energy efficiency. An IR-drop compensation scheme and a multi-bias column mapping method are proposed to address read noise (Figure 2(e)). The encryption-decryption process for privacy is experimentally implemented on eight 144 Kb 2T2R RRAM arrays [22], with small accuracy losses of 0.73% for heart disease prediction using SVM (support vector machine) and 1.9% for fashion MNIST using 4 layers of CNN. RRAM arrays are used as both VMM units and true random number generators (TRNG). An RNS-wise (residue number system) mapping method is proposed to reduce the VMM error caused by intrinsic stochasticity in RRAM devices (Figure 2(f)). RRAM arrays demonstrate the feasibility of high-performance neural signal analysis in brain-machine interfaces [23]. RRAM arrays are used to implement both a finite impulse response (FIR) filter bank and perceptron neural network in one system. It can preprocess and decoder signals in the analog domain with high efficiency, achieving a high accuracy of 93.46%.

2.3 Co-optimization of RRAM-based CIM systems

The RRAM-based CIM system can be optimized at five levels, including neural network structures, quantization methods, data flow, circuits, and devices. At the neural network algorithm level, the goal of the CIM system is to realize general neural networks. There are various algorithms, and the CIM system should maintain accuracy as a software baseline. At the CIM chip level, the precision of the device determines the bits of the network weight, and the precision of the analog-to-digital and digital-to-analog conversion circuits determines the bits of the IFMs and output feature maps (OFMs). However, the precision of RRAM devices, the precision of peripheral circuits, and the array size are limited. Therefore, the neural network requires low-bit quantization training before being mapped to the CIM chip, and the weight matrix needs to be split into multiple arrays. At the macro level, nonidealities in peripheral circuits, such as ADC quantization noise, ADC offset, and DAC noise, can induce VMM calculation errors. At the array level, interconnection resistance between devices in the array causes IR drop. This can lead to errors in the programming values and results of the VMM. At the device level, reliability issues and nonidealities of RRAM can lead to errors in the weights stored in devices, which can affect the system computing performance.

The impact of nonidealities in RRAM devices and circuits cannot be evaluated and optimized at a single abstraction layer. For example, further optimization for nonidealities in RRAM devices is difficult owing to the limitations of the device material and physical mechanism. However, optimization can be considered at the level of data flow and neural network algorithms. In addition, with the development of AI, the structure of deep neural networks is becoming increasingly complex. This increases the requirements for the design of accelerators. Therefore, the CIM system necessitates co-optimization to explore design spaces from algorithms to devices to provide design guidance.

Some co-optimization methods have been reported in our recent work. **Device-macro-algorithm co-optimization.** Owing to the nonidealities and noise of devices and circuits in CIM systems, neural networks need noise-aware offline training and low-bit quantization training to achieve high classification accuracy. During noise-aware training, nonidealities and noise are considered in the offline training process. Through the self-adaptation of neural networks, specific weights are trained for the CIM system. During low-bit quantization training, the neural network weights are quantitated according to the precision of RRAM devices, and the IFMs and OFMs are quantitated according to the precision of the ADC/DAC in the CIM system. **Device-array co-optimization.** Variations in RRAM devices are difficult to eliminate because of the inherent filament-based conductive mechanism of RRAM. An array-level boosting method is proposed to reduce the accuracy loss caused by the limited precision and noise of RRAM [24]. The array-level spatial extended allocation method can reduce variations in addition and averaging. RRAM arrays were replicated N times, and the average values of the array outputs were calculated. In addition, the greedy spatial extended allocation (GSEA) algorithm is proposed to determine the replication number N of each layer. The accuracy of ResNet-34 on the CIFAR-10 dataset with the array-level boosting method is close to the software-based accuracy (93.2%), with approximately 56% overhead of area usage and 36% of power consumption. **Array-algorithm co-optimization.** For the

IR-drop problem, the diagonal matrix regression layer (DMRL) [25] method is proposed for array optimization. It incorporates interconnect resistance effects and sneak path problems into the ex situ training of neural networks. The derived gradient of the neural network is equal to the product of the standard gradients and diagonal matrices of the DMR model [26]. **Device-macro co-optimization.** For the read noise problem, a multi-bias column mapping method is proposed [21], which uses multiple columns instead of one represented as bias. The mapping strategy is read-noise-tolerable with low energy and area overhead. **Device-chip co-optimization.** For reliability degradation, an on-chip hybrid training scheme can recover the accuracy loss [27]. This method can improve the accuracy of approaching the baseline after several iterations.

2.4 Architectural optimization: M3D architecture

Analog RRAM arrays could perform VMM with extremely high energy efficiency. However, the frequent data transfer between RRAM arrays and off-chip memory with limited bandwidth causes substantial latency and limits the parallelism of the CIM. In addition, the AD/DA area and power overhead also reduce system-level energy efficiency. Therefore, an M3D-based hybrid CIM architecture is proposed to further improve energy efficiency and parallelism [28–30]. The M3D architecture can enable the full implementation of large-scale neural networks with high efficiency [31]. The M3D chip consists of three layers: the 1st Si CMOS layer, the 2nd CIM layer of 1T1R analog RRAM arrays, and the 3rd processing-near-memory (PNM) layer based on complementary field-effect transistor (CFET) circuits with CNT/IGZO. The dense interlayer vias between layers can provide ultra-high bandwidth. Intensive VMM computations and data processing are performed by the CIM and PNM layers, respectively. An enhanced deep super-resolution (EDSR) network is implemented on the M3D chip with 149× lower energy consumption compared with the GPU. In addition, another M3D architecture of Si-based CMOS logic, RRAM-based CIM, and carbon nanotube field-effect transistor (CNTFETs)-based ternary content-addressable memory (TCAM) layers is demonstrated to implement one-shot/few-shot learning. It shows 162× lower energy consumption than the GPU [32].

3 CIM simulator

3.1 Hardware simulator

Simulator framework. A device-circuit-algorithm co-design simulator framework is built to provide guidelines for designing CIM systems [2, 33]. The RRAM-based CIM system can be optimized at five levels, including neural network structures, quantization methods, data flow, circuits, and devices. The co-design simulator considers the impact of each optimization level to realize high energy efficiency and acceptable accuracy error. As shown in Figure 3(a), the weights of the neural network and datasets are initially prepared. RRAM device parameters, such as on/off ratio, array size, and input/output precision, are configured for the simulator to parse. Then, the weights are mapped onto multiple XBs according to general matrix-matrix multiplications (GEMMs). Next, the XBs calculate the output based on the device and array models. Finally, the simulator outputs the inference accuracy and total performance of the CIM chips. A physics-based RRAM compact model is embedded in the framework (Figure 3(b)) [27, 34–36]. Nonideal effects and reliability issues are fully considered in the inference and online training processes. All the circuit modules of the XBs, as shown in Figure 3(c), are modeled for performance and accuracy evaluation [15,25–27,36]. For the XB computation, the digital inputs are coded to analog voltage levels by DACs. Then, the VMM is implemented on RRAM arrays, and the analog current outputs are quantified by ADCs. The nonidealities of peripheral circuits are taken into account in the XB analog computing model. In addition, the diagonal matrix regression (DMR) model is used to model the IR-drop effects [26]. To validate the simulator, it is calibrated with the fully hardware-implemented PCB-level multi-array integrated system and 160 kb fully-integrated CIM chip (Figure 3(d)) [5, 19, 27]. The accuracy results and circuit performance trends matched well during the hybrid training process.

Device model. The proposed physics-based RRAM compact model fully considered the nonideal effects and reliability issues of analog RRAM devices, as shown in Figure 4. The HfO_2-based analog RRAM has a material stack of TiN/TEL/HfO_2/TiN [37]. Its resistance switching is the result of the formation and rupture of conductive filaments (CFs) in the HfO_2 layer. The analog RRAM is multiple-weak-filament-based because of the percolation effect, and the oxygen vacancies (Vo) determine the

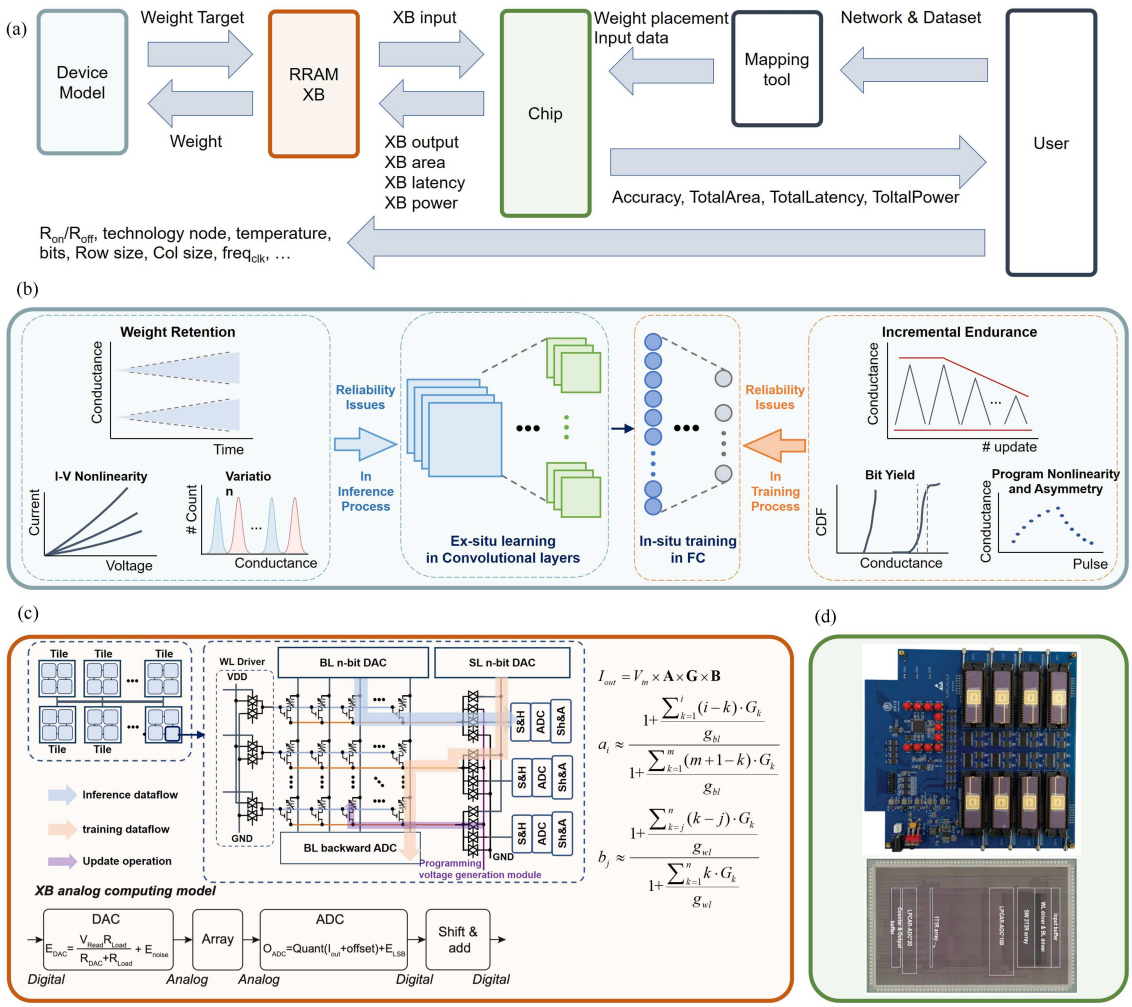


Figure 3 The simulator framework. (a) Device-architecture-algorithm co-design [2,33]; (b) embedded compact models: retention, *I-V* nonlinearity, variations affect the inference process, and endurance, bit yield, and weight update nonlinearity/asymmetry affect the online training process [27,34–36] Copyright 2021 IEEE; (c) XB analog computing model [15,25–27,36] Copyright 2021 IEEE; (d) calibration with prototype hardware systems [5,19,27] Copyright 2020 Nature and 2020 IEEE.

number of CFs and the conductivity of each CF [9,27]. The connecting state can be described by the gap length. A lower Vo density in one CF results in a longer gap length. The resistance state (RS) of CFs is similar to the electron tunneling process through the gap. For the *I-V* nonlinearity [36], a quantum point contact (QPC) model is used to describe the tunneling behavior (Eq. (1) in Figure 4). For the programming process [36], the changes in the gap length and CF numbers are modeled based on the directed percolation scheme (Eqs. (2) and (3) in Figure 4). For endurance [27], to bridge the final current states and endurance cycle number, the relationship between the final state and update numbers is presented in Eqs. (4) and (5) in Figure 4. For retention [27], in the higher RS, the gap length determines the RS, and its change obeys Brownian motion. In the lower RS, the broken CF number obeys the Poisson distribution. The probability density functions (PDF) of a higher RS and lower RS are presented in Eqs. (7) and (10) in Figure 4, respectively. The relaxation model [35] and thermal model [38] are based on the statistical measurement results (Eqs. (11)–(14) in Figure 4).

3.2 CIM compiler

For CIM chips, the dataflow can be blocked in the first several computing-intensive layers, resulting in an unbalanced utilization of tiles. To bridge the gap between the algorithm and chip, the CIM compiler is proposed to realize hardware-software co-optimization [15]. It has the ability to optimize the on-chip dataflow by reallocating hardware resources. The compiler framework is shown in Figure 5(a). The open neural network exchange (ONNX) format of the neural network is the input of the compiler.

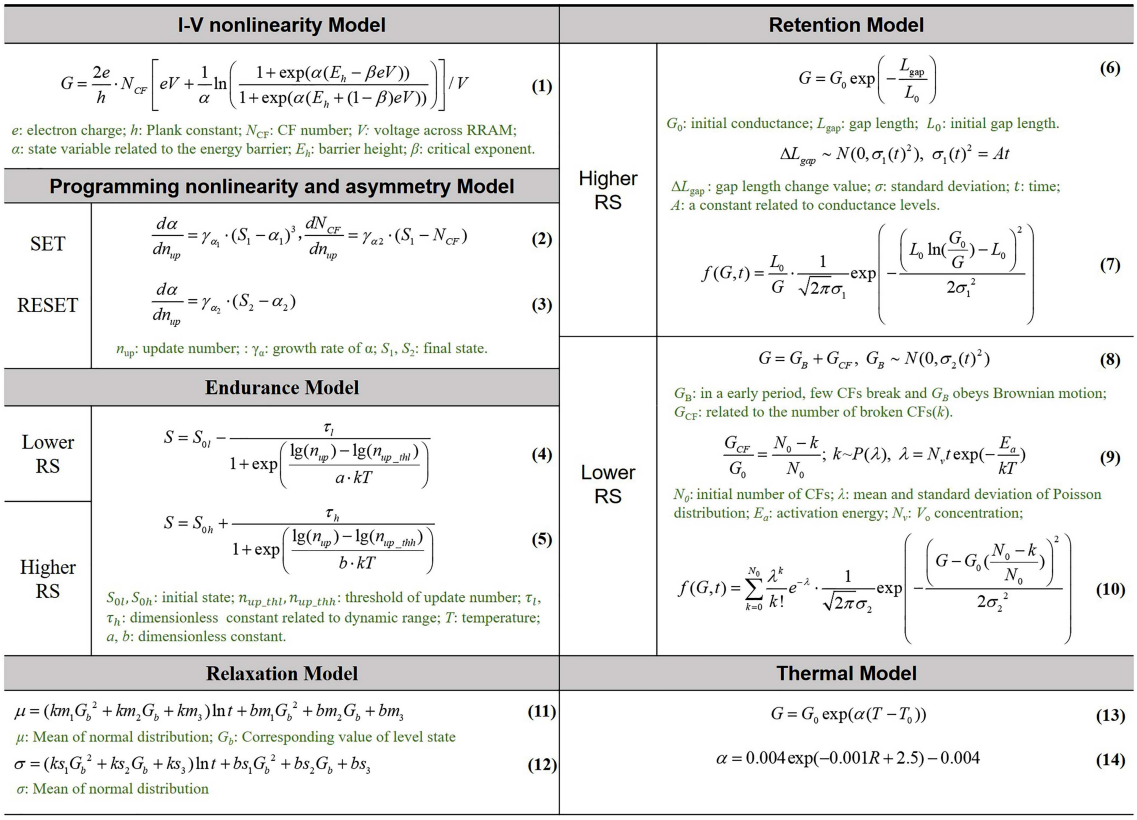


Figure 4 The analog RRAM device model considering nonidealities and reliability degradation [34] Copyright 2021 IEEE.

ONNX is parsed, analyzed, and optimized by frontend (Figure 5(b)). One optimization is the node combination, such as splitting a CONV into a combination of VMM operations and fusing the CONV and batch normalization (BN) layers into one layer. The other is critical path reforming, which entails re-allocating remaining resources to computing-intensive layers to improve throughput and utilization. Next, Frontend produces an optimized model, the intermediate representation data format file, which includes the address information and operation types for each layer in the new model. Then, the backend takes the intermediate representation as input and allocates hardware resources to the neural network layers. Finally, it produces code that can be executed on CIM chips. Figure 5(c) shows the chip-in-loop emulation. If one layer cannot be realized by real hardware, its address will be virtual, and the operation of this layer is processed with the XB analog computing model (Figure 3(c)) based on PyTorch. If the layer can be realized by real hardware, its address will be a specific location of the hardware, and VMM operations of the layer will be executed on the RRAM chip.

By adopting the proposed compilation scheme, the throughput can be significantly improved for different networks when the hardware resources are determined, as illustrated in Figure 6(a). When compared to the networks without compilation, the throughput can be improved by at least several tens of times.

This work is compared with previous studies on the simulation platform of CIM chips on five representative aspects: compilation optimization, hardware system support, inference model, training, and on-chip verification (Figure 6(b)). Given a specific hardware architecture, our compiler can find an optimized deployment. With compilation optimization, it can automatically place various AI models on hardware or simulators more reasonably. The compiler consolidated the hardware and simulator workflows, making the verification process faster and more convenient. To validate the proposed simulation model, the results of the chips and simulator were compared.

4 Future perspectives

For the CIM architecture, the weight data are stored in RRAM arrays, but the IFM and OFM data are still transmitted between the XBs and the off-chip memory. The amount of IFM and OFM data

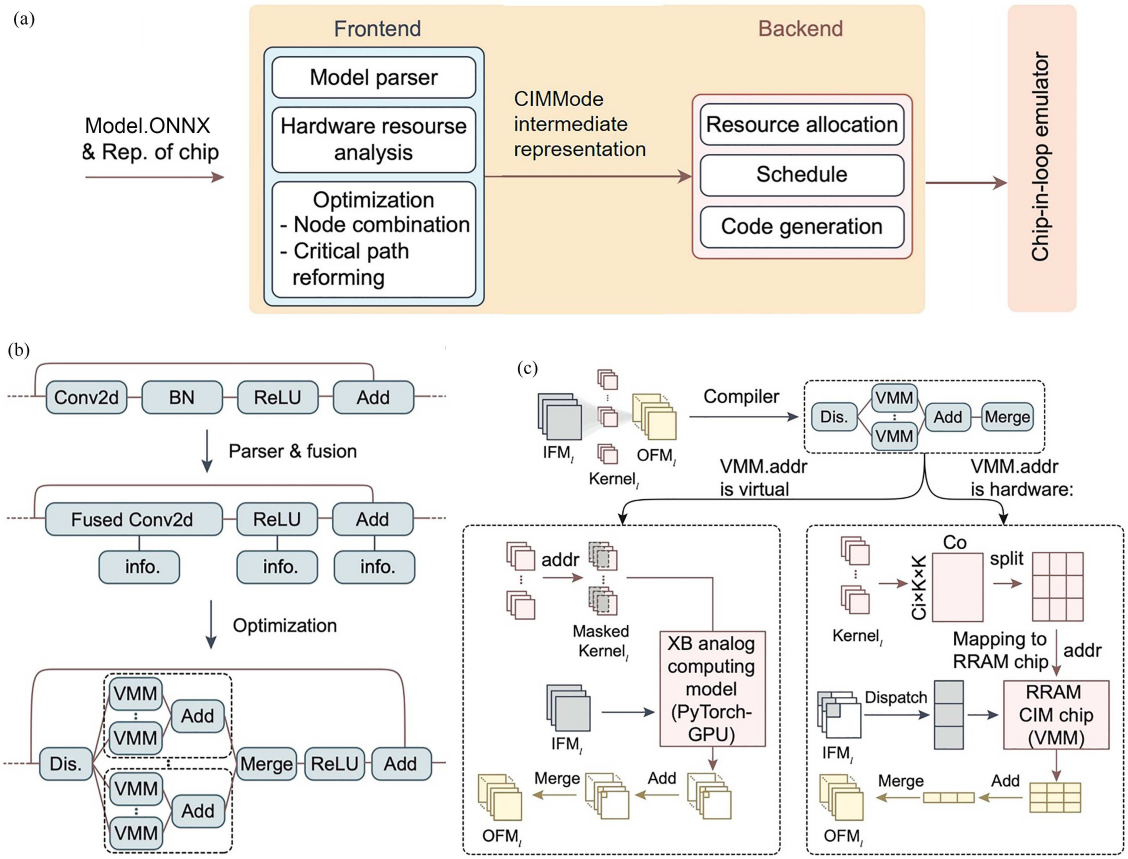
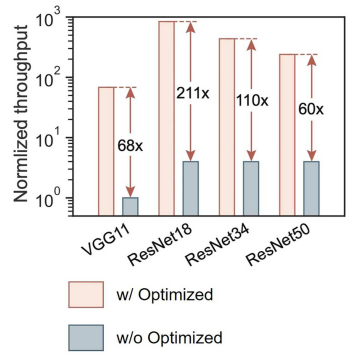


Figure 5 (a) The compiler framework; (b) workflow of the frontend in compiler; (c) chip-in-loop emulation of CIM chip [15].

(a)



(b)

Works	Compilation Optimization	Hardware System Support	Inference Model	Training	On-chip Verification
This work	√	√	Circuit Model (High Simulation Precision)	√	√
PUMAsim [39]	×	×	Behavior Model (Low Simulation Precision)	×	×
MNSIM [13]/ MNSIM2.0 [14]	×	×	Behavior Model (Low Simulation Precision)	×	×
NeuroSim [11]/ DNN+NeuroSim V2.0 [40]	×	×	Circuit Model (High Simulation Precision)	√	√

Figure 6 (a) Throughput optimization with compiler; (b) comparison with other studies [15].

is tremendous. To further eliminate the "memory wall", novel architectures need to be proposed, such as the M3D-based CIM architecture [31] and ADC-less architecture [39–41]. Furthermore, to improve scalability and generality for more applications, hybrid digital-analog accelerators with reconfigurable on-chip interconnection fabrics can be employed [42].

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 92064001, 62025111, 92264201) and Beijing Advanced Innovation Center for Integrated Circuits.

References

1 Xu X, Ding Y, Hu S X, et al. Scaling for edge inference of deep neural networks. Nat Electron, 2018, 1: 216–222
2 Zhang W Q, Gao B, Tang J S, et al. Neuro-inspired computing chips. Nat Electron, 2020, 3: 371–382

3 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

4 Zidan M A, Strachan J P, Lu W D. The future of electronics based on memristive systems. Nat Electron, 2018, 1: 22–29

5 Yao P, Wu H Q, Gao B, et al. Fully hardware-implemented memristor convolutional neural network. Nature, 2020, 577: 641–646

6 Ren Y M, Tian B B, Yan M G, et al. Associative learning of a three-terminal memristor network for digits recognition. Sci China Inf Sci, 2023, 66: 122403

7 Prezioso M, Merrikh-Bayat F, Hoskins B D, et al. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. Nature, 2015, 521: 61–64

8 Jiang Y N, Huang P, Zhou Z, et al. Circuit design of RRAM-based neuromorphic hardware systems for classification and modified Hebbian learning. Sci China Inf Sci, 2019, 62: 62408

9 Zhao M R, Gao B, Tang J S, et al. Reliability of analog resistive switching memory for neuromorphic computing. Appl Phys Rev, 2020, 7: 011301

10 Han R Z, Huang P, Zhao Y D, et al. Efficient evaluation model including interconnect resistance effect for large scale RRAM crossbar array matrix computing. Sci China Inf Sci, 2019, 62: 022401

11 Chen P Y, Peng X C, Yu S M. NeuroSim: a circuit-level macro model for benchmarking neuro-inspired architectures in online learning. IEEE Trans Comput-Aided Des Integr Circ Syst, 2018, 37: 3067–3080

12 Peng X C, Huang S S, Luo Y D, et al. DNN+NeuroSim: an end-to-end benchmarking framework for compute-in-memory accelerators with versatile device technologies. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), 2019

13 Xia L X, Li B X, Tang T Q, et al. MNSIM: simulation platform for memristor-based neuromorphic computing system. In: Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016. 469–474

14 Zhu Z H, Sun H B, Qiu K Z, et al. MNSIM 2.0: a behavior-level modeling tool for memristor-based neuromorphic computing systems. In: Proceedings of the Great Lakes Symposium on VLSI, 2020. 83–88

15 Yu R H, Zhang W Q, Gao B, et al. CLEAR: a full-stack chip-in-loop emulator for analog RRAM based computing-in-memory system. Sci China Inf Sci, 2023. doi: 10.1007/s11432-022-3756-3

16 Yao P, Wu H Q, Gao B, et al. Face classification using electronic synapses. Nat Commun, 2017, 8: 15199

17 Gao B, Zhou Y, Zhang Q T, et al. Memristor-based analogue computing for brain-inspired sound localization with in situ training. Nat Commun, 2022, 13: 2026

18 Zhou Y, Gao B, Zhang Q T, et al. Application of mathematical morphology operation with memristor-based computation-in-memory architecture for detecting manufacturing defects. Fundamental Res, 2022, 2: 123–130

19 Liu Q, Gao B, Yao P, et al. A fully integrated analog ReRAM based 78.4TOPS/W compute-in-memory chip with fully parallel MAC computing. In: Proceedings of IEEE International Solid-State Circuits Conference (ISSCC), 2020. 500–502

20 Lin Y, Hu X S, Qian H, et al. Bayesian neural network realization by exploiting inherent stochastic characteristics of analog RRAM. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), 2019

21 Qin Q, Gao B, Liu Q, et al. Hybrid precoding with a fully-parallel large-scale analog RRAM array for 5G/6G MIMO communication system. In: Proceedings of International Electron Devices Meeting (IEDM), 2022

22 Li X Q, Gao B, Lin B H, et al. First demonstration of homomorphic encryption using multi-functional RRAM arrays with a novel noise-modulation scheme. In: Proceedings of International Electron Devices Meeting (IEDM), 2022

23 Liu Z W, Tang J S, Gao B, et al. Neural signal analysis with memristor arrays towards high-efficiency brain-machine interfaces. Nat Commun, 2020, 11: 4234

24 Zhang W Q, Gao B, Yao P, et al. Array-level boosting method with spatial extended allocation to improve the accuracy of memristor based computing-in-memory chips. Sci China Inf Sci, 2021, 64: 160406

25 Liao Y, Gao B, Yao P, et al. Diagonal matrix regression layer: training neural networks on resistive crossbars with interconnect resistance effect. IEEE Trans Comput-Aided Des Integr Circ Syst, 2021, 40: 1662–1671

26 Liao Y, Gao B, Zhang W Q, et al. Parasitic resistance effect analysis in RRAM-based TCAM for memory augmented neural networks. In: Proceedings of IEEE International Memory Workshop (IMW), 2020. 1–4

27 Liu Y Y, Zhao M R, Gao B, et al. Compact reliability model of analog RRAM for computation-in-memory device-to-system codesign and benchmark. IEEE Trans Electron Dev, 2021, 68: 2686–2692

28 Shulaker M M, Hills G, Park R S, et al. Three-dimensional integration of nanotechnologies for computing and data storage on a single chip. Nature, 2017, 547: 74–78

29 Sabry A M M, Wu T F, Bartolo A, et al. The N3XT approach to energy-efficient abundant-data computing. Proc IEEE, 2019, 107: 19–48

30 Hwang W, Wan W, Mitra S, et al. Coming up N3XT, after 2D scaling of Si CMOS. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), 2018. 1–5

31 An R, Li Y J, Tang J S, et al. A hybrid computing-in-memory architecture by monolithic 3D integration of BEOL CNT/IGZO-based CFET logic and analog RRAM. In: Proceedings of International Electron Devices Meeting (IEDM), 2022

32 Li Y J, Tang J S, Gao B, et al. Monolithic 3D integration of logic, memory and computing-in-memory for one-shot learning. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), 2021

33 Zhang W Q, Peng X C, Wu H Q, et al. Design guidelines of RRAM based neural-processing-unit: a joint device-circuit-algorithm analysis. In: Proceedings of the 56th Annual Design Automation Conference, 2019. 1–6

34 Liu Y Y, Gao B. System and technology co-optimization for RRAM based computation-in-memory chip. In: Proceedings of International Conference on IC Design and Technology (ICICDT), 2021. 1–4

35 Liu Y Y, Gao B, Xu F, et al. A compact model for relaxation effect in analog RRAM for computation-in-memory system design and benchmark. In: Proceedings of the 5th IEEE Electron Devices Technology & Manufacturing Conference (EDTM), 2021. 1–3

36 Liao Y, Gao B, Xu F, et al. A compact model of analog RRAM with device and array nonideal effects for neuromorphic systems. IEEE Trans Electron Dev, 2020, 67: 1593–1599

37 Wu W, Wu H Q, Gao B, et al. A methodology to improve linearity of analog RRAM for neuromorphic computing. In: Proceedings of IEEE Symposium on VLSI Technology, 2018. 103–104

38 Ma A W, Gao B, Liu Y Y, et al. Multi-scale thermal modeling of RRAM-based 3D monolithic-integrated computing-in-memory chips. In: Proceedings of International Electron Devices Meeting (IEDM), 2022

39 Jiang H W, Huang S S, Li W T, et al. ENNA: an efficient neural network accelerator design based on ADC-free compute-in-memory subarrays. IEEE Trans Circ Syst I, 2023, 70: 353–363

40 Li W T, Xu P F, Zhao Y, et al. TIMELY: pushing data movements and interfaces in pim accelerators towards local and in time domain. In: Proceedings of the 47th Annual International Symposium on Computer Architecture (ISCA), 2020. 832–845

41 Chou T, Tang W, Botimer J, et al. CASCADE: connecting RRAMs to extend analog dataflow in an end-to-end in-memory processing paradigm. In: Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2019. 114–125

42 Ueyoshi K, Papistas I A, Houshmand P, et al. DIANA: an end-to-end energy-efficient digital and analog hybrid neural network SoC. In: Proceedings of IEEE International Solid-State Circuits Conference (ISSCC), 2022. 1–3