SCIENCE CHINA
Information Sciences

June 2021, Vol. 64 160403:1–160403:15
https://doi.org/10.1007/s11432-021-3234-0

• REVIEW •
Special Focus on Near-memory and In-memory Computing

Energy-efficient computing-in-memory architecture for AI processor: device, circuit, architecture perspective

Liang CHANG, Chenglong LI, Zhaomin ZHANG, Jianbiao XIAO, Qingsong LIU, Zhen ZHU, Weihang LI, Zixuan ZHU, Siqi YANG & Jun ZHOU*

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731*, China*

Received 8 January 2021/Revised 15 March 2021/Accepted 5 April 2021/Published online 11 May 2021

Abstract An artificial intelligence (AI) processor is a promising solution for energy-efficient data processing, including health monitoring and image/voice recognition. However, data movements between compute part and memory induce memory wall and power wall challenges to the conventional computing architecture. Recently, the memory-centric architecture has been revised to solve the data movement issue, where the memory is equipped with the compute-capable memory technique, namely, computing-in-memory (CIM). In this paper, we analyze the requirement of AI algorithms on the data movement and low power requirement of AI processors. In addition, we introduce the story of CIM and implementation methodologies of CIM architecture. Furthermore, we present several novel solutions beyond traditional analog-digital mixed static random-access memory (SRAM)-based CIM architecture. Finally, recent CIM tape-out studies are listed and discussed.

Keywords energy efficiency, computing-in-memory, non-volatile memory, test demonstrators, AI processor

Citation Chang L, Li C L, Zhang Z M, et al. Energy-efficient computing-in-memory architecture for AI processor: device, circuit, architecture perspective. Sci China Inf Sci, 2021, 64(6): 160403, https://doi.org/10.1007/s11432-021-3234-0

1 Introduction

Deep neural networks (DNNs) have become leading machine learning methods thanks to their state-of-the-art performance on various tasks, such as health monitoring and image/voice recognition [1–3]. With the development of applications on DNNs, several proposals for DNN accelerators attract attention from both academy and industry. The DianNao family provided acceleration based on the customized inner-product units with specialized instruction set architecture (ISA) [4,5]. NetFlow, Eyeriss, and TPU developed a 2D systolic array for convolutional neural networks (CNNs) considering the dataflow on the energy efficiency [6–8]. The row-stationary heuristic was analyzed to improve the performance of the accelerator by reducing data movements. These accelerators promote a novel research area on the artificial intelligent processor, namely artificial intelligence (AI) processor, which is used to accelerate the AI algorithm including but not limited to DNNs, artificial NNs (ANNs), spike NNs (SNNs).

Generally, high performance and energy-efficiency are evaluation metrics of AI processors, promoting many optimization techniques. In CNNs, there are seven-level loop nests in convolution. The process engines (PEs) of AI processors are designed to compute each loop, and transform activations to buffers and continue other loops, as shown in Figure 1. The loop computation can be reordered to improve data reuse and energy-efficiency. Consequently, the most efficient part of the CNNs algorithm can be accelerated with more PEs and fine-grained control flow. In addition, the memory buffer and dataflow choices are customized to significantly improve the energy efficiency or processing speed. However, large

* Corresponding author (email: zhouj@uestc.edu.cn)

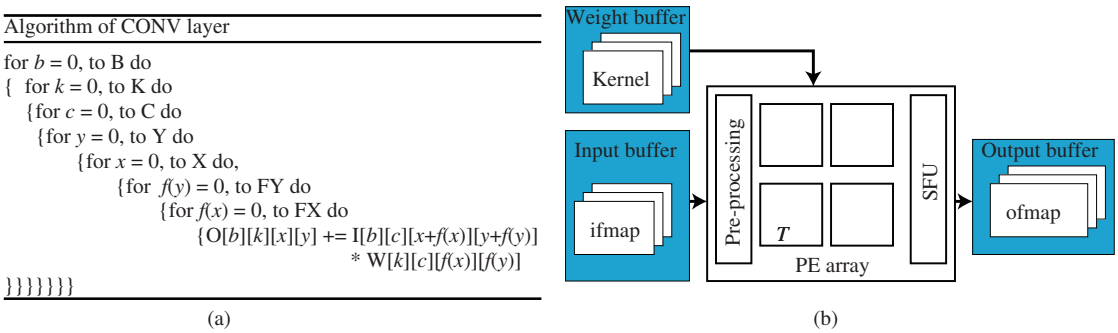

Figure 1 (Color online) The computation for the convolution layer of CNNs. (a) Simple seven nested loops of convolution layer; (b) traditional AI accelerator architecture with buffers and PE array.

data movements are required between computing logic and memory of AI processors, inducing "memory wall" and "power wall", which obstacle the development of advanced AI processors.

Recently, more and more work is focused on bridging the gap between computing logic and memory from both hardware and application sides. An effective solution is moving the computing-centric architecture to memory-centric architecture, which contains two directions. The one direction is the memory-rich computing architecture, which is equipped with high performance computing resource and large-capacity memory inside the same die. The most of execution is performed inside the chip reducing data movements as a non-Von Neumann architecture [9]. Another direction is the computing-in-memory (CIM) architecture, where the memory resource is capable of computation. The CIM computing scheme benefits its large bandwidth and decreases data movements overhead between computing logic and off-chip memory [10]. Based on our investigation of AI processors, accessing data takes two orders of magnitude energy than that of computation by PEs, leading to the memory access bottleneck and energy issue [11].

To investigate more solutions and summarize the state-of-the-art literature, this paper presents both memory-rich computing architecture and CIM architecture regarding traditional static random-access memory (SRAM), and emerging non-volatile memory (NVM). In those solutions, a large range of AI applications are employed to accelerate the computation, including artificial NNs (ANNs) and Spike NNs (SNNs). In each solution, we study characteristics of recent developments on the requirement of energy-efficiency and large data movements. We discuss the trend in the development of AI algorithms and accelerating techniques. In addition, both application-driven and technology-driven computer architecture are summarized to indicate affect factors of AI processors. Furthermore, we provide recent tape-out studies and demonstrators on SRAMs, emerging NVMs including resistive random-access memory (RRAM), magnetoresistance RAM (MRAM), and phase-change memory (PCM). The SRAM solution begins to mature and can become a product in near future, while NVM solutions may take longer to maturely using in AI processors. Through our investigation in this paper, we hope to provide a detailed history and recent developments of memory-centric computing architecture, but many solutions are still at the beginning. For instance, the MRAM and RRAM solutions are struggling in their manufacturing techniques.

The rest of this paper is organized as follows. Section 2 presents the study of CIM history, background of modern CIM technologies, and the AI-driven memory-centric computing architecture. In Section 3, the advanced memory-centric architectures are introduced, including near memory processing architecture, SRAM-based mixed signal CIM architecture, and innovations beyond mixed solution. In Section 4, we investigate the state-of-the-art literature on tape-out work of SRAM-based and NVM-based CIMs, and discuss the development of CIM architecture. We conclude this paper in Section 5.

2 Preliminary of computing-in-memory and application-driven memory-centric computing

In this section, we introduce the history of the traditional CIM technologies and architecture. Previously, the CIM was proposed to breakthrough the memory limitation by combining memory and computing logic onto the same die, as shown in Figure 2(a). This solution was failed due to the limitation of CMOS technology and application. Recently, the CIM theory is revived thanks to big data and machine learning. This section discusses energy efficiency requirements demanded by the current machine learning,

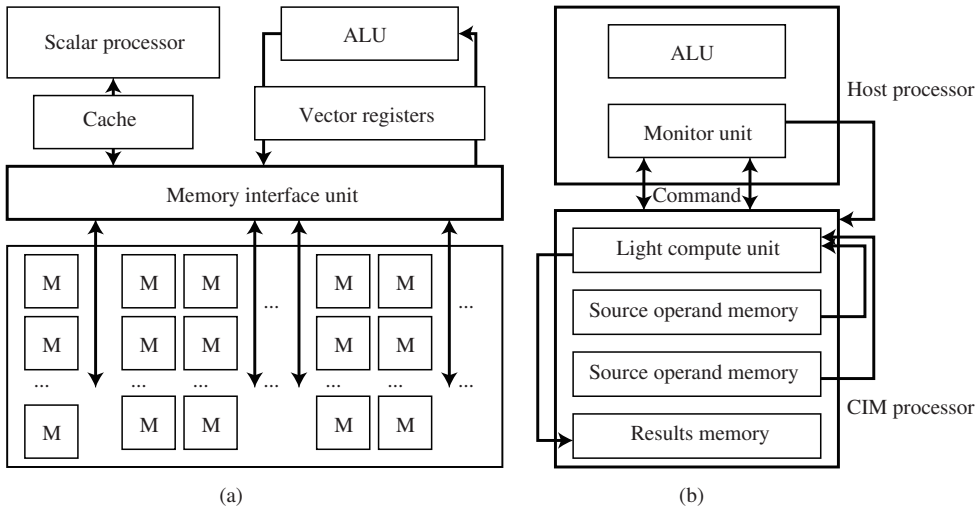

Figure 2 The CIM architecture in the 1990s and current. (a) The CIM idea of the 1990s, with compute unit nearby the memory; (b) the recent CIM architecture by integrating computing and memory.

particularly for the edge computing architecture.

2.1 History of CIMs and unsuccess of traditional CIMs

As early as in the 1990s, the idea of CIM architecture was very popular abstracting many attentions from predecessors, such as Teras, IRAM, DIVA, Active Page, FlexRAM [12–16]. In 1995, Gokhale et al. [12] proposed a Terasys system equipped with a designed and fabricated processor-in-memory processor chip. The chip is a standard 4-bit memory augmented with a single-bit arithmetic logical unit (ALU) controlling each column of memory, which can perform a local operation and optionally read/write OR operation. In 1997, Patterson et al. [17] proposed IRAM to solve the gap between processor and DRAM speed development. The IRAM merged processing and memory into a single chip to decrease the access latency of memory, increasing memory bandwidth, and improved the energy efficiency. In 1998, Oskin et al. [15] proposed active pages to implement the computation model into the SimpleScalar simulator by performing computation on reconfigurable DRAM. The evaluation results demonstrated 1000× speedups on several applications. In 1999, Hall et al. [14] proposed DIVA with computable memories connecting to one or more external host processors, performing selected computation in memory, and providing communication for moving both data and communication throughout memory. The DIVA bypassed the processor-memory bus supporting both sparse-matrix and pointer-based computations. Kang et al. [16] proposed FlexRAM to place processing-in-memory chips in the memory system to support general-purpose workstations and servers. Simulation-based evaluation illustrated 4 FlexRAM chips allow a workstation to run 25–40 times faster.

Unfortunately, the efforts of the 1990s on CIM architecture are failed to change the pattern of computer architecture for several practical concerns and limitations. First, it is difficult to integrate complex logics with DRAM technologies. Typically, there are 3 or 4 metal layers for memory, but the computing logic requires more than 10 layers in manufacturing [18]. Secondly, the optimization direction of memory prefers high density, low cost, and low leakage transistors, while optimized logic prefers high-speed transistors. Specifically, even containing simple adders in DRAM process technologies induces considerable overhead and decreases the performance of memory [19]. Third, data movement amounts are not a critical issue at that time and Moore's law still follows the rule. Also, the software and algorithm are developed slowly and the dominant application can find a suitable hardware accelerator. Moreover, technology-driven computer architecture promoted the performance of processors doubling in that years. In summary, CIM integrated with main memory is optimized for memory density and low power consumption. To merge the logic die and memory die can achieve the goal to reduce data movement between the processor and off-chip memory, but the area overhead of memory causes server concern.

Recently, with the development of complementary metal-oxide-semiconductor transistor (CMOS) and memory technologies, the previous approach on main memory has been revisited, such as buffered comparator, DRISA, and SCOPE [20–22]. Several novel implementations on the CIM architecture have

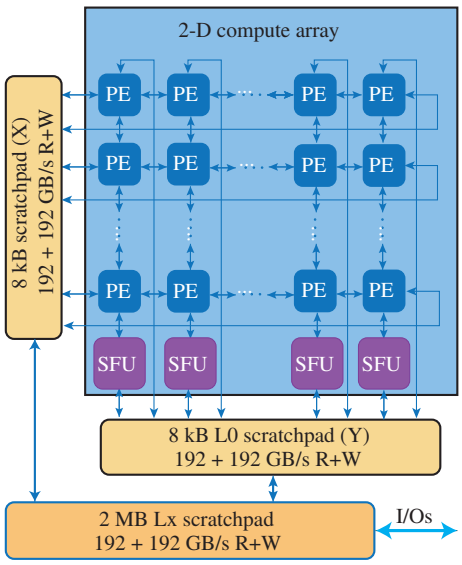

Figure 3 (Color online) An example of AI-driven architecture with PE array, scratchpad memory for both AI training and inference [29].

been developed. The dominant workloads, including machine learning algorithms, big data processing, cloud computing, and internet-of-thing, are emerging to drive the hardware architecture and circuit to search on revolutionary solutions [23–27]. In addition, the time comes to the post-Moore era, and the technology-driven benefit of computer architecture is decline. Novel non-von Neumann architecture is anticipated to solve the data-movement problem and a tremendous improvement on the performance of AI processor is desired. Therefore, we review various recent technologies to design the CIM architecture regarding SRAM and emerging NVMs. We analyze the advantages and disadvantages of each technology to find the possible direction of post-Moore computing architecture. Based on aforementioned concerns and limitations, we discuss various memory technologies to demonstrate possibilities of memory-centric computing architecture.

2.2 AI-driven memory-centric computing

Recently, deep CNNs (DCNNs) have been used in many applications to provide high performance such as accuracy, speed, energy-efficiency. However, DNNs are characterized by intensive computation and dependence on cascade memory accesses. In DCNNs, the high computational complexity that comes from simultaneously processing hundreds of filters and channels in the high-dimensional convolutions, is an obstacle for the corresponding AI processor. In particular, to large NNs, filter weights and feature maps are stored in main memory. In the computing process, the accelerator will read filter weights and bias only once, but may read and write feature maps several times. Various techniques are employed to optimize the DNNs regarding algorithm and hardware accelerators. Chen et al. [28] categorized the data movements in a spatial architecture into several levels of hierarchy according to their energy cost, and then analyzed each dataflow to assess the data movement at each level. They provided four levels of storage hierarchy, including DRAM, global buffer, the memory array of inter-PE communication, and register file. Eyeriss achieves high energy-efficiency and speed by the systolic array and memory development. As shown in Figure 3 [29], although a highly parallel computing array can meet the need of the computation requirement, the significant data movements induced energy consumption can be higher than that of computation. For considering the layer-wise activation maps, the significant data movements in hardware accelerators can operate on real-time streaming inputs. Based on the above analyses, Samal et al. [30] explored an architecture-aware algorithmic approach to reduce data movement and the resulting latency and power consumption. They presented attention-based feedback for controlling input data, referred to as the activation pruning, that reduces activation maps in early layers of a DNN network which are critical for reducing data movement in real-time AI processing. Experiments on sequences from the KITTI dataset show the activation pruning maintains quality of motion planning while increasing the sparsity of the activation maps.

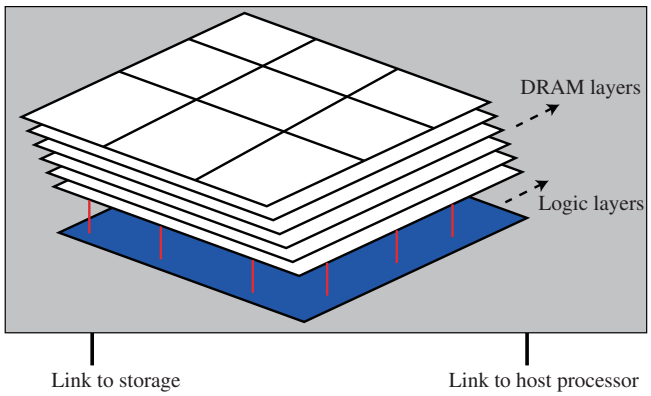

Figure 4 (Color online) The architecture of near memory computing based on 3D stacking technology.

For embedded computer vision in portable systems, real-time processing is required within a limited power budget. The specialized memory-centric hardware architecture can be designed to meet the intensive computation and memory access issues. Based on the memory-centric optimization, Yin et al. [31] proposed an affine computation architecture, using specialized pipeline design, which greatly improves computing efficiency. Consequently, the AI-driven memory-centric computing architecture can improve the performance of AI processors. In this work, we summarize recent literature on memory-centric computing architecture to provide more solutions for energy-efficient AI processors.

3 Advanced memory-centric architecture for intelligent computing

This section discusses advanced memory-centric architectures including both near memory processing architecture and CIM architecture. We introduce the theory of several novel schemes for the recent CIM architecture to indicate the possible design with mature technologies. In addition, we present the SRAM mixed signal processing for the CIM, LUT-based CIM organization, and possible time-domain based computing strategy.

3.1 Near memory processing architecture for the intelligent computing

Near memory processing (NMP) architecture can be realized by the 3D storage technique to provide high bandwidth. There are two types of successful 3D stack memory. One is the high-bandwidth memory (HBM) of JEDEC [32]. Another is the hybrid memory cube (HMC) from Micron [33]. Both HBM and HMC contain a single chip stacked with multiple DRAM devices, as shown in Figure 4.

Typically, the memory hierarchy of computer architecture consists of cache, main memory, and disk. The raw data should move from disk all the way to the cache, inducing a gap between computing unit and memory. To solve the memory access issue, the NMC processes raw data near the data. For example, the HMC contains one logic layer to process the data inside HMC without moving data outside the HMC chip, which reduces latency and energy of datamovements. This principle can be applied to novel computing architecture, such as on-board buffer processing, edge bound small processor chip with DRAM chip [26]. Ahn et al. [23] developed an effective communication mechanism between different cube cores based on message passing, namely TESSERACT. In TESSERACT, the long-time remote access delay was hidden by using non-blocking message passing, and atomic memory can be updated without software synchronization primitives, hence to fully use the memory bandwidth of a simple core. Koo et al. [34] proposed SUMMARIZER, where the inherent ARM core was located inside an SSD combined as a processor. The data intensive task offloads to the SSD-based processor to accelerate the processing. Nair et al. [35] developed AMC upon the HMC, in which several processing elements are added onto HMA working as lanes in the logic layer. The host processor and each AMC are interconnected over a bidirectional link following a daisy-chain topology. This study provides an interconnection topology for the NMC solution. Farmahini-Farahani et al. [36] proposed three different NMC architectures using coarse-grained reconfigurable arrays on commodity DRAM modules. This proposal requires minimal change to the DRAM architecture. However, a programmer should identify which code will run close to

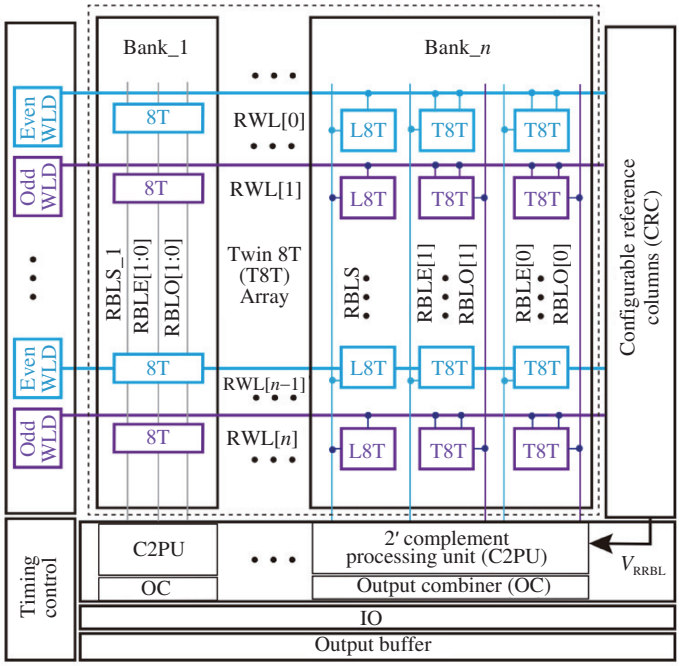

Figure 5 (Color online) The modified memory cell based SRAM CIM architecture [37].

memory, leading to more programmer efforts for compute intensive code executing on the 3D stack logic layer, see Figure 5 [37].

Recently, the NMC is also employed to accelerate graph processing, including GraphP, GraphH, GraphQ [19, 38, 39]. GraphP uses PIM based graph processing system to greatly reduce energy consumption by developing the hardware and software cooperative interface. GraphP provides a partition technique to change the communication between cross cube, designs a programming model to maintain the partition, and proposes a layer communication/overlapping strategy to further improve performance. Compared to TESSERACT, GraphP provides on average 1.7× speedups and 89% energy saving [38]. GraphH integrates SRAM-based vertex buffer to eliminate local bandwidth degradation. Moreover, GraphH introduces a reconfigurable dual grid connection to provide high global bandwidth, and provides mapping and partition scheduling methods to achieve workload balance and conflict avoidance. Based on the evaluation, GraphH achieves up to a 5.12× speedup compared with Tesseract [39]. GraphQ implements batch and overlapping cube communication by reordering vertex processing. In addition, the heterogeneous cores are used for different access types to simplify the communication between cubes. Furthermore, a hybrid execution model is proposed to perform additional local computation. Based on the simulation, GraphQ achieves on average 3.3× and maximum 13.9× speedup, 81% energy saving compared to Tesseract.

3.2 SRAM-based analog-digital mixed CIM architecture

Recently, CIM design with SRAM is mature gradually thanks to the traditional SRAM technology. As for the mixed signal processing in CIM architectures, two typical ways are commonly used to convert multi-bit inputs into WL signals. One is the fully parallel input structure [40]. The parallel charging and discharging operation of bit lines can be described in the following process. First, the bit line pair (BL/BLB) is precharged. Then, the analog voltage representing the eigenvalue is used to drive the WLS to generate the corresponding bit cell current. Each current applied to BL or BLB depends on the data stored in the bit cell. Therefore, taking BL/BLB as the differential voltage signal, it can be considered that the weight of the stored data multiplied by an eigenvalue is -1 or $+1$ respectively. Finally, the current of all bit cells is added on BL/BLB, resulting in aggregation discharge, and the comparator provides a symbol threshold. Another is the fully serial input structure [41]. In the architecture given in this paper, if 8-bit ADC is used as a readout circuit to detect bit line current, it will cause large-area power overhead [42]. Therefore, a 1-bit comparator is proposed to replace the ADC. In order to make positive BL (PBL) and negative BL (NBL) have the same current, multi bits are input in order, and PBL

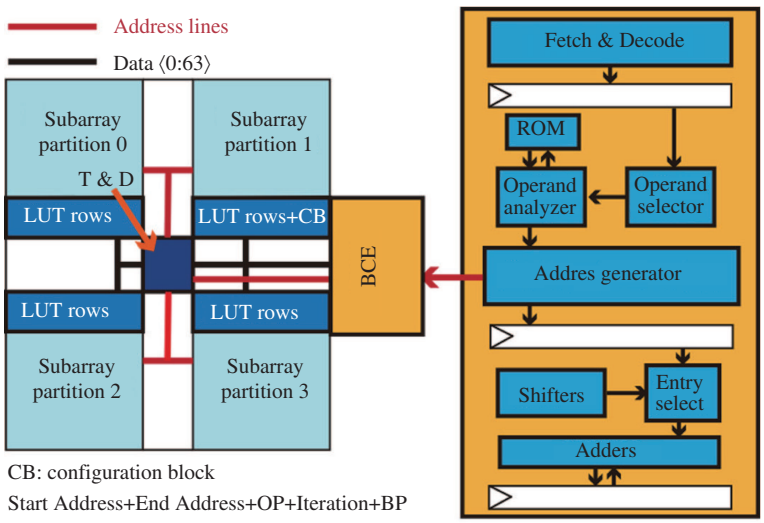

Figure 6 (Color online) The look-up-table based CIM architecture [47].

and NBL are compared through bit serial by controlling CD signal, the number of R1 units is changed in each step.

Kang et al. [43] presented an integrated circuit implementation of a random forest (RF) machine learning classifier based on SRAM. To reduce the complexity of interconnection and avoid irregular memory access, this work used deterministic subsampling and balanced decision tree. By providing massively parallel processing, Kang et al. implemented low swing analog computing in memory embedded in standard 6T SRAM. The prototype IC achieves a 3.1× energy savings and 2.2× speed-up at the same time providing a 6.8× lower energy-delay product (EDP) at the same accuracy of larger than 93% compared to conventional digital architecture. Valavi et al. [44] presented a method to minimize weight movement and eliminate multiple output activations in NN accelerators. In 2018, Kang et al. developed prototype adopting the deep memory architecture, namely DIMA. The DIMA embedded the low swing analog processing of column in peripheral of multi-row for standard 6T bit-cell array at the same time via each precharge operation. Compared to traditional digital architecture, the DIMA improved energy efficiency and throughput [45]. In 2019, Valavi et al. proposed a 65 nm prototype designed with Mixed signal CIM, which considered the data flow of machine learning algorithm and used charge domain mixed signal operation to manage data movements. This design also achieves high energy efficiency and throughput executing convolution layers. Then, to design a robust memory supporting vector machine classifier, the on-chip training technique was used to decrease PVT influence [46]. In 2020, Si et al. [37] developed a configurable twin-8T (T8T) SRAM-CIM unit-macro with 1–4-bit input precision, 1–5-bit weight precision and up to 7-bit output precision for various multibit DCNN applications.

3.3 CIM architecture beyond mixed solution

In the analog-digital mixed CIM architecture, there are some concerns. First, the ADC is required to perform the computation, which occupies a large area and is sensitive to process, voltage, temperature conditions. In addition, the verification of large scale mixed signal-based SoC chips is difficult to implement since the accuracy prototype model is difficult to develop. In this subsection, we discuss the advanced CIM architecture beyond the mixed solution, including look-up table based CIM architecture (see Figure 6 [47]) and time-domain based computing scheme.

The look-up table (LUT)-based CIM architecture is proposed to solve the issue of mixed-based CIM solution [47]. In LUT-based CIM, two memory rows are designed for LUT, supporting two modes: storage and computation modes. In the storage mode, the LUT array is separated from the memory array, reducing additional power consumption. In the computation mode, LUT-based CIM provides three operations: multiplication, division, and activation by caching the key data inside LUT. In the multiplication operation, the LUT-CIM converts the even number of computation into a shift operation, reducing the required size of LUT from 225 to 49 bits. For matrix multiplication, the steps of multiplication, judgment and accumulation are reorganized as multilevel pipelines to improve the parallelism of the architecture. In dividing operation, the hard-to-calculate $\frac{1}{\text{dividend}}$ value is located into the LUT to save resources and

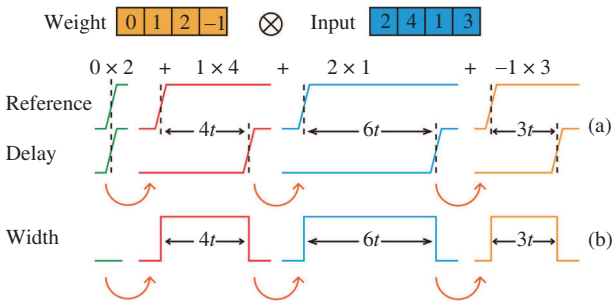

Figure 7 (Color online) The possible time-domain based CIM theory.

energy consumption, avoiding designing complicated division hardware unit. For the activation function, the piecewise linear approximation method is used to store the value of each iteration in the LUT, and calculate the final value of functions including counting and accumulating the partial sum. Based on the evaluation, the proposed LUT-based CIM architecture achieves 1.72× overall speedup and 3.14× energy saving for the L3 cache compared to the neural cache [48].

Another alternative solution is time-domain computation with CIM architecture. In the traditional digital domain of multiplication and addition, large energy consumption of switching capacitors is wasted. Time-domain computing is a method to implement multiplication and addition using the width or delayed superposition of pulses [49–52]. As shown in Figure 7, a common time-domain calculation consists of two delay lines: reference line and delay line. The calculation can be implemented with a rising edge delay, and different inputs and weights are delayed and advanced the rising edge in the delay line by the corresponding time value. Then, the final results of multiplication and addition can be achieved by the time interval between the rising edge of the delay line and the reference line. Although the time domain is a part of the analog domain, its effective values are only 0 and 1, which can be calculated as a digital domain in nature. Unlike voltage-domain calculations that require large analog-to-digital conversions, the conversion from the time domain to the digital domain can be achieved simply by a counter or a reference line delay detection [49–52]. The energy-efficient time-domain computing is suitable for low-power AI processors. In 2018, Amravati et al. [53, 54] used time-domain computing in the field of reinforcement learning and proposed an energy efficient framework for computing time-domain mixed signals. It is compatible with multi-bit weighting operations and uses a numerical control oscillator to implement time-domain multiplication and addition. In 2019, Sayal et al. [50] proposed a convolutional neural network computational engine for all-digital time-domain computing. A novel bidirectional memory delay line unit is adopted to perform signed accumulation of the input and weighted product, and the conversion from the time domain to the digital domain is achieved using a simple counter. By adjusting the accuracy of the digital to time conversion, the calculation engine also enables four modes of acceleration, which to a certain extent alleviates the problem of slow computing caused by the cumulative nature of the time domain delay. In the same year, Chen and Gu [55] designed a time-domain computational gas pedal for the dynamic time-regularization algorithm, a classical algorithm for time series classification problems. To implement the pipeline, they designed a special time domain flip-flop as the time domain memory. The pipelined operation is then implemented by this time domain flip-flop circuit, which significantly improves the performance and scalability of the operation. In 2020, Li et al. [9] further proposed an innovative ReRAM-CIM accelerator by combining time-domain computing, non-volatile memory, and CIM architecture. By using the time-domain interface and the used analog local buffer separately, the energy of each digital time conversion and the total number of conversions are greatly reduced.

3.4 Emerging non-volatile memory solutions on CIM architecture

Another promising solution for the CIM architecture is using the emerging NVMs including RRAM, phase-change random access memory (PCRAM), and MRAM [56–58], as shown in Figure 8 [59]. The RRAM device is a metal-insulator-metal structure, where a metal-oxide layer is sandwiched between metal electrodes as switching material. The high resistance state (HRS) and low resistance state (LRS) can be switched when the amplitude of external voltage exceeds the threshold [56]. The resistance of RRAM cell could vary from serval KΩ to tens of MΩ. PCRAM device exploits chalcogenide glass as a switching element. Chalcogenide glass is a group of fragile glass switching between the amorphous

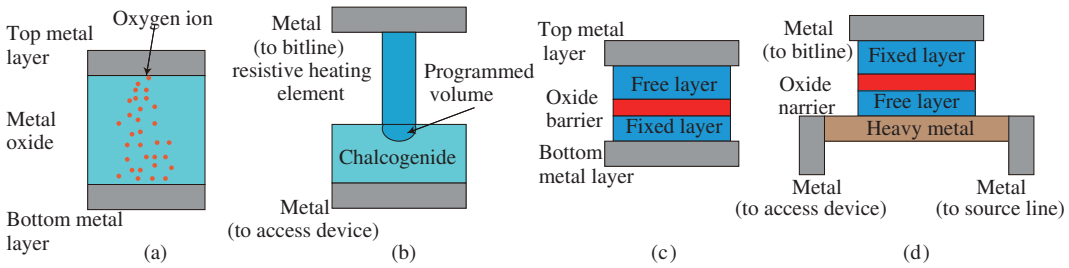

Figure 8 (Color online) The emerging NVM device contains (a) resistive RAM, (b) phase-change RAM, (c) spin-transfer torque MRAM, and (d) spin-orbit torque MRAM [59].

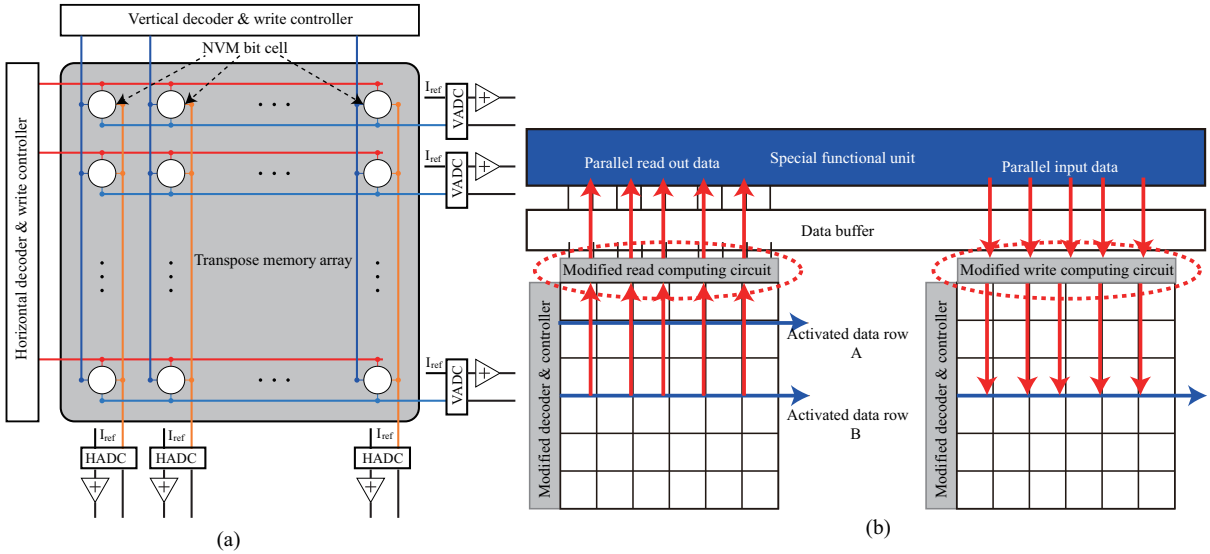

Figure 9 (Color online) The CIM architecture with NVM device. (a) Transpose memory array; (b) CIM architecture supports bit-wise operations.

state and crystalline state by current controlled heating and cooling [57]. Amorphous state chalcogenide glass can provide a resistance over 1 MΩ resulting in HRS of PCRAM cell while crystalline state glass provides only up to 10 kΩ resistance pulling PCRAM cell into LRS. Magnetic tunnel junction (MTJ) is the primary storage core of MRAM, which contains the free layer, oxide barrier, reference layer [58]. The magnetization orientation of free layer can be changed to parallel and anti-parallel states, which represents low resistance and high resistance. For programing the MRAM, there are two mainly MTJ structures including spin transfer torque MTJ as shown in Figure 8(c), and spin orbit torque MTJ as shown in Figure 8(d). Compared to the traditional SRAM, NVMs have demonstrated low leakage power, high density, more energy-efficiency advantages [60].

NVM-based CIM architecture is designed for several commonly used mathematical calculations, such as MAC operation, matrix transposition, bit-wise logic operations. Conventional CIM macro only uses two states: HRS and LRS [10]. For MAC operation implemented by NVM-based CIM, more than one memory rows are activated and each memory cell would contribute a part of total bitline current, converting to voltages compared to different reference cells. Then the sense amplifier can recognize the computation result. In the NN acceleration, filter weights are stored in memory cells, and feature maps are converted into SL voltage for each corresponding word line. The MAC results can be parallel computed to improve the performance. In addition, the multi-states resistive cell can provide higher precision and density to the CIM architecture [61]. For instance, the resistance of RRAM cell can be changed continuously from KΩ to MΩ between HRS and LRS [62–64]. The RRAM-based CIM solution employs an analog-digital converter to identify the different values, hence to implement high-precision MAC operation. The NVMs can also be used to perform the transpose operation with additional wires and switches. Typically, the memory array can be read and wrote row by row controlled by the decoder. With simple modification on memory column and peripheral circuit, the memory array can support fast transpose operations used in the NN architecture. A typical architecture of NVM based CIM PE for MAC and transposition acceleration is displayed in Figure 9(a) [27, 65]. For bit-wise logic operations, both input features and

Table 1 Summarization of modern CIM test demonstrations

Publication	Technology (cell)	Target (Operation)	Capacity	Method	Precision	Accuracy (%)
2017 [40]	SRAM 130 nm (6T)	SVM (-)[a)]	16 kb	Mixed	I(5b), W(1b), O(1b)	MNIST: 90
2018 [42]	SRAM 65 nm (10T)	CNN (F)	4 kb	Mixed	I(7b), W(1b), O(4b)	MNIST: 96
2018 [66]	SRAM 65 nm (6T)	DNN (F)	4 kb	Mixed	I(1b), W(1b), O(1b)	MNIST: 97.5
2019 [37]	SRAM 65 nm (T8T)	CNN (F)	3.8 kb	Mixed	I(4b), W(5b), O(7b)[b)]	MNIST: 99.52
2020 [67]	SRAM 28 nm (6T)	CNN (F/B)	64 kb	Mixed	I(8b), W(8b), O(20b)[b)]	CIFAR: 91.94
2020 [68]	SRAM 7 nm (8T)	CNN (F)	4 kb	Mixed	I(4b), W(4b), O(4b)	MNIST: 98.50
2020 [69]	SRAM 28 nm (6T)	CNN (F)	64 kb	Mixed	I(8b), W(8b), O(20b)[b)]	CIFAR10: 92.02
2020 [70]	SRAM 65 nm (8T)	CNN (F)	4 kb	Mixed	I(4b), W(8b), O(20b)[c)]	RsNet: 92.88
2019 [71]	SRAM 28 nm (8T)	DSP (-)	128 kb	Mixed	I(-), W(-), O(-)	–
2018 [72]	SRAM 65 nm (8T)	SVM (F)	16 kb	Mixed	I(8b), W(8b), O(-)[c)]	MIT-CBCL: 96
2020 [73]	SRAM 55 nm (6T)	CNN (F)	4 kb	Mixed	I(8b), W(8b), O(19b)[b)]	CIFAR10: 91.93
2019 [74]	SRAM 28 nm (8T)	arithmetic (-)	16 kb	Mixed	I(-), W(-), O(-)[b)]	–
2020 [75]	SRAM 65 nm (8T)	CNN (F)	72 kb	Mixed	I(8b), W(4b), O(-)	MNIST: 92.40
2020 [76]	SRAM 65 nm (8T1C)	CNN (F)	72 kb	Mixed	I(1b), W(1b), O(5b)	MNIST: 98.30
2020 [77]	SRAM 65 nm (12T)	DNN (F)	16 kb	Mixed	I(1b), W(3b), O(-)	MNIST: 98.84
2019 [78]	SRAM 65 nm (10T)	CNN (F)	16 kb	Mixed	I(6b), W(1b), O(6b)	MNIST: 98.30
2019 [45]	SRAM 65 nm (6T)	SVM (-)[a)]	16 kb	Mixed	I(8b), W(8b), O(-)	MIT: 95
2019 [79]	SRAM 28 nm (8T)	CNN (F)	2 kb	Time domain	I(8b), W(1b), O(8b)	–
2021 [80]	SRAM 28 nm (10T)	MAC (F)	2.6 Mb	All digital	I(4 or 8b), W(4–16b), O(all)	–
2018 [81]	RRAM 65 nm (1T1R)	CNN/FCN (F)[d)]	1 Mb	NVM	I(1b), W(3b), O(3b)	MNIST: 98
2019 [82]	RRAM 55 nm (1T1R)	CNN (F)	1 Mb	NVM	I(2b), W(3b), O(3b)	MNIST: 98.80
2020 [83]	RRAM 150 nm (1T1R)	CNN (F)	64 kb	NVM	I(8b), W(3b), O(-)	CIFAR10:98.90
2020 [84]	RRAM 150 nm (1T1R)	CNN (F)	–	NVM	I(-), W(-), O(-)	–
2020 [85]	RRAM 130 nm (2T2R)	MAC (F)	158.8 kb	NVM	I(1b), W(3b), O(8b)	MNIST: 94.40
2020 [56]	RRAM 130 nm (1T1R)	RNN/CNN/MLP (F)[c)]	500 kb	NVM	I(1b),W(-),O(1b)	MNIST: 97.55
2020 [86]	RRAM 55 nm(1T1R)	CNN (F)	1 Mb	NVM	I(2b), W(3b), O(4b)	CIFAR10: 88.52
2020 [87]	RRAM 130 nm (1T1R)	CNN/BNN (F)	16 kb	NVM	I(-), W(-), O(-)	–
2020 [88]	RRAM 130 nm (1T1R)	MVM (F)	64 kb	NVM	I(2b), W(3b), O(-)	MINIST: 91.38
2017 [89]	PCRAM-(1T1R)	Stat.(-)	3 Mb	NVM	–	Statistic: 93
2020 [90]	PCRAM-(1T1R)	DNN (F)	256 kb	NVM	I(8b), W(8b), O(8b)	CIFAR10: 93.7
2020 [60]	STT-MRAM (1T1R)	SHA (-)	1 Mb	NVM	–	–
2020 [91]	STT-MRAM (1T1R)	CNN (F)	8 Mb	NVM	I(8b), W(8b), O(8b)	MIT-BIH: 85.1

a) They supported k-NN/SVM/MF/TM algorithm and executed MIT CBCL, GUN shot sound, MNIST dataset.
b) They supported 3 precision types, and we state the maximum precision and accuracy.
c) They supported 3 operations including forward, reverse, recurrent.
d) They supported 3 precision types, and it is a system and computes layer by layer.

weights are located inside the memory array, as demonstrated in Figure 9(b). For the computation stage, two or more memory rows are activated to read-out the results [18,25]. The write-based computation also can be supported by the NVM-based solution by activating both one memory row or multi-rows [10,59].

4 Test demonstrators and discussion

This section provides a summarization of the recent tape-out work on CIM architecture, such as SRAM-, RRAM-, MRAM- and PCRAM-based CIM architectures. Based on our investigation, we found that the most of studies are on SRAM and RRAM-based CIM test chips. Only a few tape-out studies are on MRAM and PCRAM-based chips. The detailed information is listed in Table 1 [37,40,42,45,56,60,66–91], including technology, implementation method, targe application, capacity, precision and accuracy.

4.1 Test prototypes and demonstrators

In the beginning, processors with CIM architecture can only handle some simple logical operations. In 2016, Jeloka et al. [92] designed a CIM macro based on 6T SRAM that can perform bit-wise AND and OR operations with 1b precision. In 2017, Dong et al. [68] proposed a CIM macro with Boolean logic functions (AND, OR, XOR) between the two activated inputs. One step closer, the CIM processors

of traditional algorithm base on vector product operation were proposed one after another. In 2017, Zhang et al. [40] provided a VCM processor with 1b precision, achieving more than 90% accuracy on the MNIST data set. In 2018, Gonugondla et al. [72] designed a low energy consumption VCM processor edge processing system, which reached 8b precision and 96% accuracy on the MIT-CBCL data set. The same year, Kang et al. [45] designed a multi-functional CIM inference processor with 8b precision, which supports four algorithms, and all of them have an accuracy of over 92%. In 2019, Wang et al. [74] proposed a general-purpose hybrid in and near-memory with unlimited precision input, which can perform Boolean, Integer arithmetic, float arithmetic. In addition, the excellent energy-saving and acceleration performance is provided by CIM architecture, promoting AI processors towards the DCNN algorithm. In 2017, Ando et al. [93] designed a binary/ternary deep neural networks reconfigurable accelerator based on CIM architecture, which achieving energy efficiency that 102–104 times better than a CPU/GPU/FPGA. In 2017, Su et al. [84] proposed an RRAM-Based FCNN energy-saving processor, which achieves 462 GOPs/J energy efficiency. Chen et al. [81] proposed an RRAM-Based macro of binary CNN, which achieves a 98% MNIST recognition rate. Zhang et al. [40] proposed an SRAM-Based macro of binary/ternary DNN, which achieves a 98.3% MNIST recognition rate.

Most of the early work on CIM can only handle low precision networks. Lately, CIM AI edge chips have implemented multi-bit precision for input, weight, and output data, which is sufficient for practical applications. In 2019, Biswas et al. [78] presented an energy-efficient SRAM with 6-bit inputs/outputs, which achieves 40.3 TOPS/W energy efficiency and a 98.3% MNIST recognition rate. In 2020, Si et al. [37] developed a T8T SRAM-CIM macro with 1–4-bit inputs, 1–5-bit weights, and 7-bit outputs for multi-bit CNN, which achieves the highest 99.02% TOPS/W energy efficiency and a 90.4% CIFAR10 recognition rate. Xue et al. [86] designed a 2 Mb ReRAM-CIM macro and improved the input precision to 4 bit. Further, achieve a 9.8–18.3 ns latency and an energy efficiency up to 121.3 TOPS/W. In Table 1, the most common CIM memory array is SRAM and RRAM. In 2016, Kang et al. [45] used a standard 6T SRAM cell to implement a 4 kb CIM macro with 1-bit logic operations. In 2018, Yin et al. [77] designed a CIM macro based on 12T SRAM that can perform XNOR operations for binary networks. In 2020, Si et al. [37] designed a Twin-8T SRAM bit cell, which completed 2-bit weighted multiplication and port isolation with 1.58× area cost (compare with 6T SRAM). Wang et al. [71] designed an 8T SRAM cell, which transposes the bit data to read to achieve the purpose of maintaining compatibility with the mainstream CPU/GPU architecture. Jiang et al. [76] designed an 8T1C SRAM cell to perform capacitive-coupling, which isolates the influence of calculation on bit cell. In 2019, Yang et al. [79] designed a memory computing chip based on 8T SRAM. In this design, the pulse width modulation unit is used to realize MAC operation in the time domain. A single-bit weighted neural network accelerator with 8b input, 1b weight, and 8b output accuracy is realized, and the energy efficiency ratio of 46.6 TOPS/W is achieved. Recently, Chih et al. [80] proposed a 10T SRAM-based all-digital CIM macro for multiply-accumulate (MAC) operations. It can support 1–8-bit input activations, 4–16-bit weights for full precision calculation with the configurable signed. And the energy efficiency ratio reached 89 TOPS/W.

In the RRAM-based CIM macro design, 1T1R is the most common structure. In the past few years, researchers have implemented a variety of CIM chips with different bit precision and application scenarios based on 1T1R [56, 81–84, 86–88]. In 2020, Liu et al. [85] proposed a sign-weighted 2T2R bit cell, which can output the positive and negative weight on the same column and eventually boost the computing parallelism. Besides, some potential CIM cells are being studied, Slesazeck et al. [94] used the 2TnC FeRAM to achieve specific Logic budgets with 1b&2b input. Yu et al. [95] designed a 4T2C eDRAM, which can store a ternary state using four transistors only. For the NVM technology beyond RRAM, PCRAM and MRAM were used to design CIM architecture. In 2017, Sebastian et al. [89] of the IBM Research Center successfully ran unsupervised machine learning algorithms on one million PCM devices, which is expected to increase speed and energy efficiency by 200 times. In 2020, IBM proposed a training method of ResNet convolutional neural network based on PCRAM CIM structure, which significantly improved the accuracy of weight transfer. To solve the error problem of resistive memory, a compensation technique is proposed to obtain the highest accuracy based on PCRAM memory on CIFAR-10 (93.7%) and ImageNet (71.6%) data sets. In 2020, Chang et al. [60] designed a 1 Mb STT-MRAM Macro in 22 nm technology, which reduced 3 times power consumption compared with a similar design, and achieved the readout bandwidth of 42.67 Gb/s. In the same year, Lee et al. [91] proposed an STT-MRAM-based ECG arrhythmia monitoring system. Combined with the characteristics of non-volatile memory and time-domain computing, the ultra-low total power consumption of 1.02 μW was achieved in near memory architecture.

4.2 Discussion on the development of CIM

Potentially, the CIM is becoming more and more complex with supporting various calculations and operations. The optimization direction for modern CIM architecture is developed to high precision, high throughput, and high energy efficiency. The implemented methods are varied from analog-digital mixed signal processing, all-digital, time-domain, and NVMs, respectively. For the mixed-based solution, ADCs are employed to read the final results with the analog computing method, which is sensitive to technology process, voltage, and temperature. In addition, the area-efficiency is limited by the high precision ADC since the area of current ADCs is large. Furthermore, the accurate validation and prototype verification are difficult with mixed-based CIM macro in a large scale SoC chip. The all digital SRAM-based CIM architectures provide a compromise solution to develop high precision, and stable CIM architecture for AI processors. Also, the time-domain based solution is another alternative to avoid process-voltage-temperature (PVT) issues and suitable to energy-efficient AI processors. However, the natural problem of SRAM, including low density, high leakage power consumption, low extensibility, is the obstacle to large-scale use of SRAM into AI processor. As the CMOS technology scaling down, the issue is still existent and difficult to solve from circuit and architecture levels.

NVM-based solutions become optimistic choices. Among those emerging NVM-based CIM architecture, RRAM is most used to design the CIM test chip thanks to the high resistance and low cost. The multi-level RRAM device, as memristor, is a promising characteristic for high precision CIM architecture, which can achieve high density, high parallelism, and high energy efficiency. Another memristor choice is PCRAM, supported by several research groups. MRAM has the best performance regarding latency and access energy, but the resistance and TMR value limit the usage on CIM architecture. Also, emerging NVMs are still in their infant stages with technologies, stability, scalability challenges. There is still much work to be done, such as device technology, circuit and architecture innovations. More prototypes should be tested to validate the possibility of NVM technologies, including memory array, memory structure with a controller, NVM-based CIM chips and processors.

Based on our investigation, we understand the development of the state-of-the-art solution for modern CIM in terms of device, circuit and architecture. In future work, several fronts are worth researching. First, the stability of analog-digital mixed CIM solution can be improved by fine-grained design from stand cell, compiler, and layout modifications. Second, all-digital SRAM CIM macro is designed to perform MAC operation. To employ the macro into large-scale chips should be studied both from academy and industry to check the feasibility and potentiality. Third, novel physical theories and phenomena of NVM devices can be found with involving more excellent researchers and international foundries. Innovations on the computing strategy can be developed with current NVM devices, such as LUT and the time domain. Finally, another mixed solution is the SRAM mixed with NVMs, which is a possible solution to solve both precision and stability problems. Overall, CIM, as a memory-centric computing architecture, is on the way to realization. To investigate more solutions and detailed CIM models is our future work, which beyond the scope of this article.

5 Conclusion

Both technology and application drive the development of CIM architecture, which can be deployed into CPU and AI processor. Various AI applications required large dataset and input data for the computation, which induces pressure for the interface between the computing unit and memory. In this paper, we analyzed the history of CIM architecture and presented the recent developments of modern CIM architectures. We listed the recent tape-out work on CIM chips to show the state-of-the-art literature of the SRAM-based and NVM-based CIMs. In the post-Moore age, the SRAM-based CIM solution can be a potential choice for AI processors. In the future, the NVMs may become promising alternatives as CIM-based AI processors.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2019YFB2204500) and UESTC Research Start-up Funding (Grant No. Y030202059018052).

References

1 Liu L, Qu Z, Deng L, et al. Duet: boosting deep neural network efficiency on dual-module architecture. In: Proceedings of the 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2020. 738–750
2 Wess M, Manoj P D S, Jantsch A. Neural network based ECG anomaly detection on FPGA and trade-off analysis. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), 2017. 1–4

3 Zairi H, Talha M K, Meddah K, et al. FPGA-based system for artificial neural network arrhythmia classification. Neural Comput Appl, 2019, 32: 4105–4120
4 Chen Y, Luo T, Liu S, et al. Dadiannao: a machine-learning supercomputer. In: Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture, 2014. 609–622
5 Du Z, Fasthuber R, Chen T, et al. Shidiannao: shifting vision processing closer to the sensor. In: Proceedings of ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA), 2015. 92–104
6 Pham P, Jelaca D, Farabet C, et al. Neuflow: dataflow vision processing system-on-a-chip. In: Proceedings of IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), 2012. 1044–1047
7 Chen Y, Krishna T, Emer J, et al. 14.5 eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks. In: Proceedings of IEEE International Solid-State Circuits Conference (ISSCC), 2016. 262–263
8 Jouppi N, Young C, Patil N, et al. In-datacenter performance analysis of a tensor processing unit. In: Proceedings of ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA), 2017
9 Li W, Xu P, Zhao Y, et al. Timely: pushing data movements and interfaces in PIM accelerators towards local and in time domain. In: Proceedings of ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), 2020. 832–845
10 Chi P, Li S, Xu C, et al. Prime: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In: Proceedings of ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), 2016. 27–39
11 Zhao Y, Chen X, Wang Y, et al. Smartexchange: trading higher-cost memory storage/access for lower-cost computation. In: Proceedings of ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), 2020. 954–967
12 Gokhale M, Holmes B, Iobst K. Processing in memory: the Terasys massively parallel PIM array. Computer, 1995, 28: 23–31
13 Patterson D, Anderson T, Cardwell N, et al. A case for intelligent RAM. IEEE Micro, 1997, 17: 34–44
14 Hall M, Kogge P, Koller J, et al. Mapping irregular applications to diva, a PIM-based data-intensive architecture. In: Proceedings of the ACM/IEEE Conference on Supercomputing, 1999. 57
15 Oskin M, Chong F T, Sherwood T. Active pages: a computation model for intelligent memory. In: Proceedings of the 25th Annual International Symposium on Computer Architecture, 1998. 192–203
16 Kang Y, Huang W, Yoo S M, et al. FlexRAM: toward an advanced intelligent memory system. In: Proceedings of IEEE International Conference on Computer Design, 1999. 192–201
17 Patterson D, Anderson T, Cardwell N, et al. Intelligent RAM (IRAM): chips that remember and compute. In: Proceedings of IEEE International Solids-State Circuits Conference, 1997. 224–225
18 Li S, Xu C, Zou Q, et al. Pinatubo: a processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories. In: Proceedings of the 53rd ACM/EDAC/IEEE Design Automation Conference (DAC), 2016. 1–6
19 Zhuo Y W, Wang C, Zhang M X, et al. Graphq: scalable PIM-based graph processing. In: Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture. New York: Association for Computing Machinery, 2019
20 Deng L, Wang G, Li G, et al. Tianjic: a unified and scalable chip bridging spike-based and continuous neural computation. IEEE J Solid-State Circ, 2020, 55: 2228–2246
21 Li S, Niu D, Malladi K T, et al. Drisa: a DRAM-based reconfigurable in-situ accelerator. In: Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2017. 288–301
22 Li S, Glova A O, Hu X, et al. Scope: a stochastic computing engine for DRAM-based in-situ accelerator. In: Proceedings of the 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2018. 696–709
23 Ahn J, Hong S, Yoo S, et al. A scalable processing-in-memory accelerator for parallel graph processing. In: Proceedings of ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA), 2015. 105–117
24 Chang L, Ma X, Wang Z, et al. CORN: in-buffer computing for binary neural network. In: Proceedings of Design, Automation Test in Europe Conference Exhibition (DATE), 2019. 384–389
25 Chang L, Ma X, Wang Z, et al. PXNOR-BNN: in/with spin-orbit Torque MRAM preset-XNOR operation-based binary neural networks. IEEE Trans VLSI Syst, 2019, 27: 2668–2679
26 Gao M, Ayers G, Kozyrakis C. Practical near-data processing for in-memory analytics frameworks. In: Proceedings of International Conference on Parallel Architecture and Compilation (PACT), 2015. 113–124
27 Peng X, Liu R, Yu S. Optimizing weight mapping and data flow for convolutional neural networks on processing-in-memory architectures. IEEE Trans Circ Syst I, 2020, 67: 1333–1343
28 Chen Y, Emer J, Sze V. Eyeriss: a spatial architecture for energy-efficient dataflow for convolutional neural networks. In: Proceedings of ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), 2016. 367–379
29 Fleischer B, Shukla S, Ziegler M, et al. A scalable multi-TeraOPS deep learning processor core for AI trainina and inference. In: Proceedings of IEEE Symposium on VLSI Circuits, 2018. 35–36
30 Samal K, Wolf M, Mukhopadhyay S. Attention-based activation pruning to reduce data movement in real-time AI: a case-study on local motion planning in autonomous vehicles. IEEE J Emerg Sel Top Circ Syst, 2020, 10: 306–319
31 Yin S, Ouyang P, Liu L, et al. A fast and power-efficient memory-centric architecture for affine computation. IEEE Trans Circ Syst II, 2016, 63: 668–672
32 JEDEC. High Bandwidth Memory (HBM) DRAM. JESD235A-2015. https://www.jedec.org/standards-documents/docs/jesd235a
33 Consortium H M C. Hybrid memory cube specification 1.0. 2013. https://yumpu.b4your.com/en/pdf/3015151532/
34 Koo G, Matam K K, Te I, et al. Summarizer: trading communication with computing near storage. In: Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2017. 219–231
35 Nair R, Antao S F, Bertolli C, et al. Active memory cube: a processing-in-memory architecture for exascale systems. IBM J Res Dev, 2015, 59: 1–14
36 Farmahini-Farahani A, Ahn J H, Morrow K, et al. NDA: near-DRAM acceleration architecture leveraging commodity DRAM devices and standard memory modules. In: Proceedings of IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), 2015. 283–295
37 Si X, Chen J, Tu Y, et al. A Twin-8T SRAM computation-in-memory unit-macro for multibit CNN-based AI edge processors. IEEE J Solid-State Circ, 2020, 55: 189–202
38 Zhang M, Zhuo Y, Wang C, et al. Graphp: reducing communication for PIM-based graph processing with efficient data partition. In: Proceedings of IEEE International Symposium on High Performance Computer Architecture (HPCA), 2018. 544–557
39 Dai G, Huang T, Chi Y, et al. GraphH: a processing-in-memory architecture for large-scale graph processing. IEEE Trans

Comput-Aided Des Integr Circ Syst, 2019, 38: 640–653
40 Zhang J, Wang Z, Verma N. In-memory computation of a machine-learning classifier in a standard 6T SRAM array. IEEE J Solid-State Circ, 2017, 52: 915–924
41 Okumura S, Yabuuchi M, Hijioka K, et al. A ternary based bit scalable, 8.80 TOPS/W CNN accelerator with many-core processing-in-memory architecture with 896K synapses/mm^2. In: Proceedings of Symposium on VLSI Technology, 2019
42 Biswas A, Chandrakasan A P. CONV-RAM: an energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications. In: Proceedings of IEEE International Solid-State Circuits Conference, 2018. 488–490
43 Kang M, Gonugondla S K, Shanbhag N R. A 19.4 nJ/decision 364 K decisions/s in-memory random forest classifier in 6T SRAM array. In: Proceedings of the 43rd IEEE European Solid State Circuits Conference, 2017. 263–266
44 Valavi H, Ramadge P J, Nestler E, et al. A mixed-signal binarized convolutional-neural-network accelerator integrating dense weight storage and multiplication for reduced data movement. In: Proceedings of IEEE Symposium on VLSI Circuits, 2018. 141–142
45 Kang M, Gonugondla S K, Patil A, et al. A multi-functional in-memory inference processor using a standard 6T SRAM array. IEEE J Solid-State Circ, 2018, 53: 642–655
46 Gonugondla S K, Kang M, Shanbhag N. A 42 PJ/decision 3.12 TOPS/W robust in-memory machine learning classifier with on-chip training. In: Proceedings of IEEE International Solid-State Circuits Conference, 2018. 490–492
47 Ramanathan A K, Kalsi G S, Srinivasa S, et al. Look-up table based energy efficient processing in cache support for neural network acceleration. In: Proceedings of the 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2020. 88–101
48 Eckert C, Wang X, Wang J, et al. Neural cache: bit-serial in-cache acceleration of deep neural networks. In: Proceedings of ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), 2018. 383–396
49 Sayal A, Fathima S, Nibhanupudi S S T, et al. 14.4 all-digital time-domain CNN engine using bidirectional memory delay lines for energy-efficient edge computing. In: Proceedings of IEEE International Solid-State Circuits Conference, 2019. 228–230
50 Sayal A, Nibhanupudi S S T, Fathima S, et al. A 12.08-TOPS/W all-digital time-domain CNN engine using bi-directional memory delay lines for energy efficient edge computing. IEEE J Solid-State Circ, 2020, 55: 60–75
51 Everson L R, Liu M, Pande N, et al. A 104.8 TOPS/W one-shot time-based neuromorphic chip employing dynamic threshold error correction in 65 nm. In: Proceedings of IEEE Asian Solid-State Circuits Conference (A-SSCC), 2018. 273–276
52 Everson L R, Liu M, Pande N, et al. An energy-efficient one-shot time-based neural network accelerator employing dynamic threshold error correction in 65 nm. IEEE J Solid-State Circ, 2019, 54: 2777–2785
53 Amravati A, Nasir S B, Thangadurai S, et al. A 55 nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots. In: Proceedings of IEEE International Solid-State Circuits Conference, 2018. 124–126
54 Amaravati A, Nasir S B, Ting J, et al. A 55-nm, 1.0–0.4 V, 1.25-pJ/MAC time-domain mixed-signal neuromorphic accelerator with stochastic synapses for reinforcement learning in autonomous mobile robots. IEEE J Solid-State Circ, 2019, 54: 75–87
55 Chen Z, Gu J. High-throughput dynamic time warping accelerator for time-series classification with pipelined mixed-signal time-domain computing. IEEE J Solid-State Circ, 2021, 56: 624–635
56 Wan W, Kubendran R, Eryilmaz S B, et al. 33.1 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, 2020. 498–500
57 Khwa W, Chang M, Wu J, et al. 7.3 a resistance-drift compensation scheme to reduce MLC PCM raw BER by over 100× for storage-class memory applications. In: Proceedings of IEEE International Solid-State Circuits Conference (ISSCC), 2016. 134–135
58 Wang Z, Zhou H, Wang M, et al. Proposal of toggle spin torques magnetic RAM for ultrafast computing. IEEE Electron Device Lett, 2019, 40: 726–729
59 Chang L, Ma X, Wang Z, et al. DASM: data-streaming-based computing in nonvolatile memory architecture for embedded system. IEEE Trans VLSI Syst, 2019, 27: 2046–2059
60 Chang T, Chiu Y, Lee C, et al. 13.4 a 22 nm 1 Mb 1024b-read and near-memory-computing dual-mode STT-MRAM macro with 42.6 GB/s read bandwidth for security-aware mobile devices. In: Proceedings of IEEE International Solid-State Circuits Conference, 2020. 224–226
61 Zhang S, Huang K, Shen H. A robust 8-bit non-volatile computing-in-memory core for low-power parallel MAC operations. IEEE Trans Circ Syst I, 2020, 67: 1867–1880
62 Yu Z, Wang Z, Kang J, et al. Early-stage fluctuation in low-power analog resistive memory: impacts on neural network and mitigation approach. IEEE Electron Device Lett, 2020, 41: 940–943
63 Yang J, Zhu J, Dang B, et al. TaOx synapse array based on ion profile engineering for high accuracy neuromorpic computing. In: Proceedings of China Semiconductor Technology International Conference (CSTIC), 2020. 1–4
64 Wang Z, Kang J, Bai G, et al. Self-selective resistive device with hybrid switching mode for passive crossbar memory application. IEEE Electron Device Lett, 2020, 41: 1009–1012
65 Chang L, Wang Z, Zhang Y, et al. Multi-port 1R1W transpose magnetic random access memory by hierarchical bit-line switching. IEEE Access, 2019, 7: 110463
66 Khwa W, Chen J, Li J, et al. A 65 nm 4 kb algorithm-dependent computing-in-memory SRAM unit-macro with 2.3 ns and 55.8 TOPS/W fully parallel product-sum operation for binary DNN edge processors. In: Proceedings of IEEE International Solid-State Circuits Conference, 2018. 496–498
67 Su J, Si X, Chou Y, et al. 15.2 a 28 nm 64 kb inference-training two-way transpose multibit 6T SRAM compute-in-memory macro for AI edge chips. In: Proceedings of IEEE International Solid-State Circuits Conference, 2020. 240–242
68 Dong Q, Sinangil M E, Erbagci B, et al. 15.3 a 351 TOPS/W and 372.4GOPS compute-in-memory SRAM macro in 7 nm FinFet CMOS for machine-learning applications. In: Proceedings of IEEE International Solid-State Circuits Conference, 2020. 242–244
69 Si X, Tu Y, Huang W, et al. 15.5 a 28 nm 64 kb 6T SRAM computing-in-memory macro with 8b MAC operation for AI edge chips. In: Proceedings of IEEE International Solid-State Circuits Conference, 2020. 246–248
70 Yue J, Yuan Z, Feng X, et al. 14.3 a 65 nm computing-in-memory-based CNN processor with 2.9-to-35.8 TOPS/W system energy efficiency using dynamic-sparsity performance-scaling architecture and energy-efficient inter/intra-macro data reuse. In: Proceedings of IEEE International Solid-State Circuits Conference, 2020. 234–236
71 Wang J, Wang X, Eckert C, et al. 14.2 a compute SRAM with bit-serial integer/floating-point operations for programmable

in-memory vector acceleration. In: Proceedings of IEEE International Solid-State Circuits Conference, 2019. 224–226
72 Gonugondla S K, Kang M, Shanbhag N. A 42 PJ/decision 3.12 TOPS/W robust in-memory machine learning classifier with on-chip training. In: Proceedings of IEEE International Solid-State Circuits Conference, 2018. 490–492
73 Chiu Y C, Zhang Z, Chen J J, et al. A 4-kb 1-to-8-bit configurable 6T SRAM-based computation-in-memory unit-macro for CNN-based AI edge processors. IEEE J Solid-State Circ, 2020, 55: 2790–2801
74 Wang J, Wang X, Eckert C, et al. A 28-nm compute SRAM with bit-serial logic/arithmetic operations for programmable in-memory vector computing. IEEE J Solid-State Circ, 2020, 55: 76–86
75 Jia H, Valavi H, Tang Y, et al. A programmable heterogeneous microprocessor based on bit-scalable in-memory computing. IEEE J Solid-State Circ, 2020, 55: 2609–2621
76 Jiang Z, Yin S, Seo J S, et al. C3SRAM: an in-memory-computing SRAM macro based on robust capacitive coupling computing mechanism. IEEE J Solid-State Circ, 2020, 55: 1888–1897
77 Yin S, Jiang Z, Seo J, et al. XNOR-SRAM: in-memory computing SRAM macro for binary/ternary deep neural networks. IEEE J Solid-State Circ, 2020, 55: 1733–1743
78 Biswas A, Chandrakasan A P. CONV-SRAM: an energy-efficient SRAM with in-memory dot-product computation for low-power convolutional neural networks. IEEE J Solid-State Circ, 2019, 54: 217–230
79 Yang J, Kong Y, Wang Z, et al. 24.4 sandwich-RAM: an energy-efficient in-memory BWN architecture with pulse-width modulation. In: Proceedings of IEEE International Solid-State Circuits Conference, 2019. 394–396
80 Chih Y D, Lee P H, Fujiwara H, et al. An 89 TOPS/W and 16.3 TOPS/mm2 all-digital SRAM-based full-precision compute-in memory macro in 22 nm for machine-learning edge applications. In: Proceedings of IEEE International Solid-State Circuits Conference (ISSCC), 2021. 252–254
81 Chen W, Li K, Lin W, et al. A 65 nm 1 Mb nonvolatile computing-in-memory ReRAM macro with sub-16 ns multiply-and-accumulate for binary DNN AI edge processors. In: Proceedings of IEEE International Solid-State Circuits Conference, 2018. 494–496
82 Xue C, Chen W, Liu J, et al. 24.1 a 1 Mb multibit ReRAM computing-in-memory macro with 14.6 ns parallel MAC computing time for CNN based AI edge processors. In: Proceedings of IEEE International Solid- State Circuits Conference, 2019. 388–390
83 Yan B, Yang Q, Chen W, et al. RRAM-based spiking nonvolatile computing-in-memory processing engine with precision-configurable in situ nonlinear activation. In: Proceedings of Symposium on VLSI Technology, 2019. 86–87
84 Su F, Chen W, Xia L, et al. A 462 GOPS/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, 2017. 260–261
85 Liu Q, Gao B, Yao P, et al. 33.2 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, 2020. 500–502
86 Xue C, Chen W, Liu J, et al. Embedded 1-Mb ReRAM-based computing-in- memory macro with multibit input and weight for CNN-based AI edge processors. IEEE J Solid-State Circ, 2020, 55: 203–215
87 Zha Y, Nowak E, Li J. Liquid silicon: a nonvolatile fully programmable processing-in-memory processor with monolithically integrated ReRAM. IEEE J Solid-State Circ, 2020, 55: 908–919
88 Wan W, Kubendran R, Gao B, et al. A voltage-mode sensing scheme with differential-row weight mapping for energy-efficient RRAM-based in-memory computing. In: Proceedings of IEEE Symposium on VLSI Technology, 2020. 1–2
89 Sebastian A, Tuma T, Papandreou N, et al. Temporal correlation detection using computational phase-change memory. Nature Commun, 2017, 8: 1–10
90 Joshi V, Gallo M L, Haefeli S, et al. Accurate deep neural network inference using computational phase-change memory. Nature Commun, 2020, 11: 1–13
91 Lee K R, Kim J, Kim C, et al. A 1.02-UW STT-MRAM-based DNN ECG arrhythmia monitoring SOC with leakage-based delay MAC unit. IEEE Solid-State Circ Lett, 2020, 3: 390–393
92 Jeloka S, Akesh N B, Sylvester D, et al. A 28 nm configurable memory (TCAM/BCAM/SRAM) using push-rule 6T bit cell enabling logic-in-memory. IEEE J Solid-State Circ, 2016, 51: 1009–1021
93 Ando K, Ueyoshi K, Orimo K, et al. Brein memory: a 13-layer 4.2 k neuron/0.8 m synapse binary/ternary reconfigurable in-memory deep neural network accelerator in 65 nm CMOS. In: Proceedings of Symposium on VLSI Circuits, 2017. 24–25
94 Slesazeck S, Ravsher T, Havel V, et al. A 2TnC ferroelectric memory gain cell suitable for compute-in-memory and neuromorphic application. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), 2019. 1–4
95 Yu C, Yoo T, Kim H, et al. A logic-compatible eDRAM compute-in-memory with embedded ADCs for processing neural networks. IEEE Trans Circ Syst I, 2021, 68: 667–679