IEEE Open Journal of the
Solid-State Circuits Society

Received 4 July 2021; revised 18 September 2021; accepted 20 October 2021. Date of publication 26 October 2021; date of current version 8 November 2021.

Digital Object Identifier 10.1109/OJSSCS.2021.3123287

Challenges and Trends of Nonvolatile In-Memory-Computation Circuits for AI Edge Devices

JE-MIN HUNG, CHUAN-JIA JHANG (Graduate Student Member, IEEE), PING-CHUN WU (Graduate Student Member, IEEE), YEN-CHENG CHIU, AND MENG-FAN CHANG (Fellow, IEEE)

(Invited Paper)

Institute of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan
CORRESPONDING AUTHOR: M.-F. CHANG (e-mail: mfchang@ee.nthu.edu.tw)
This work was supported by the Ministry of Science and Technology (MOST), Taiwan.

ABSTRACT Nonvolatile memory (NVM)-based computing-in-memory (nvCIM) is a promising candidate for artificial intelligence (AI) edge devices to overcome the latency and energy consumption imposed by the movement of data between memory and processors under the von Neumann architecture. This paper explores the background and basic approaches to nvCIM implementation, including input methodologies, weight formation and placement, and readout and quantization methods. This paper outlines the major challenges in the further development of nvCIM macros and reviews trends in recent silicon-verified devices.

INDEX TERMS Artificial intelligence, nonvolatile-memory, NVM, computation-in-memory, CIM, nvCIM.

I. INTRODUCTION

ADVANCED edge devices for artificial intelligence (AI) and AI enabled Internet-of-Things (AIoT) applications commonly adopt nonvolatile memory (NVM) for power-off data storage to suppress power consumption in standby mode. Most AI edge devices must perform multiply-and-accumulate (MAC) operations with high input (IN), weight (W), and output (OUT) bit-precision to achieve the inference accuracy, computing speeds, and energy consumption required for most practical applications.

Typical edge devices based on the von Neumann architecture have the processing elements (PEs) separated from the memory devices, such that PEs must frequently access data from NVM (off-chip or on-chip) via the memory bus, as shown in Fig. 1(a). Considerable read latency, high parasitic load on the data bus, and limited bandwidth for memory access in movement of data from NVM to PEs following system wake up greatly increase the overall latency and energy consumption. These effects are particularly evident when implemented in complex neural networks using large volumes of high bit-precision data. This bottleneck is known as the memory wall.

Nonvolatile on-chip computing-in-memory (nvCIM) [1]–[39], [47]–[62], [80], [81] is able to avoid the memory-wall bottleneck, and thereby reduce latency and energy consumption during power-on and AI computing operations, by enabling highly parallel computing within the memory macros to minimize the movement of intermediate data, as shown in Fig. 1(b).

Compact nvCIM devices provide high input, weight, and output bit-precision with low energy consumption, short latency, and robust readout against process variations, all of which are particularly important when dealing with complex datasets, such as CIFAR-100 [2], [3], [34] and ImageNet [5]. Volatile CIM using static random-access memory (SRAM) -based CIM and capacitive CIM have also been proposed and demonstrated to achieve the high energy efficiency and broke the von Neumann bottleneck. Capacitive CIM can be implemented using established technologies without the need for high-voltage drivers or FETs. The use of charge re-distribution allows capacitive CIM to perform MAC operations with short read-write latency.

Numerous nvCIM macros have been developed using resistive random-access memory (ReRAM) [1]–[39], [81], phase change memory (PCM) [54]–[62], [80], and magnetoresistive random-access memory (MRAM) [47]–[53] to perform MAC operations with various degrees of precision.

The remainder of the paper is organized as follows. Section II introduces the background and basic circuit structure of in-memory computation (IMC)-based nvCIM circuits

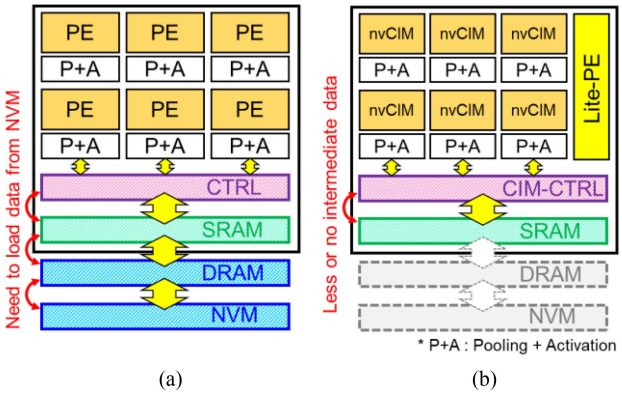

FIGURE 1. (a) Conventional von Neumann architecture; (b) NVM-based computing-in-memory (nvCIM) architecture.

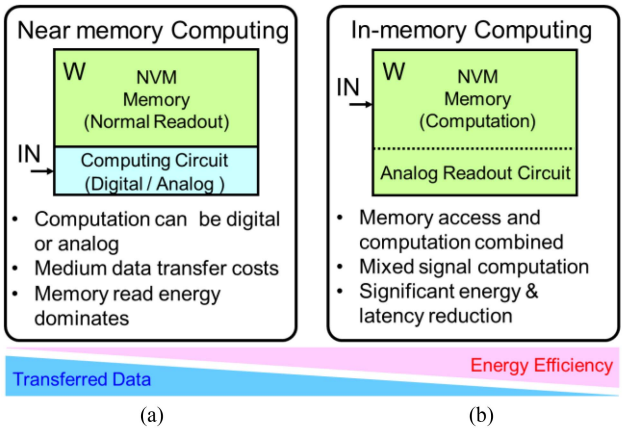

FIGURE 2. Conceptual illustrations of (a) NVM near-memory computation and (b) NVM in-memory computation.

for AI applications, including input methods, weight mapping, and readout quantization. Section III explores the major challenges involved in the further development of this technology. Section IV examined recent silicon-verified nvCIM macros from the perspective of performance and design-space analysis. Conclusions are outlined in Section V.

II. BACKGROUND AND STRUCTURES OF NVCIM

A. FUNCTIONALITY AND FEATURES OF NVCIM

In terms of functionality, nvCIM is able to perform logic operations, pattern matching computation (content address memory; CAM) [63]–[70], neuromorphic computing for spike-timing-dependent-plasticity (STDP) [77]–[78], and MAC operations [2]–[5], [17], [19], [21], [32]–[35], [79], [80]. MAC operations have become a major focus of research for intelligent edge devices, due to its importance in convolutional neural networks (CNNs), which are commonly used for deep-learning based AI devices. MAC operations involve the multiplication of inputs (IN) and weights (W) followed by the accumulation of multiplication results. When using nvCIM for MAC computing, weight data is generally stored in the NVM cell array, and inputs are processed in selected wordlines (WL) or bitlines (BL) within the NVM cell array.

The capacity of most nvCIM devices usually exceeds the mega-bit (Mb) level, which is sufficient to store all of the weight-data for the neural network models implemented in tiny AI edge devices. By training weight data in the cloud, nvCIM macros are able to focus on inference operations. The fact that they seldom perform write operation for the updating of weights and parameters means that the limited endurance of NVM devices is seldom an impediment.

B. NVCIM STRUCTURE FOR MAC OPERATIONS: NEAR-MEMORY COMPUTING (NMC) AND IN-MEMORY COMPUTING (IMC)

CIM macros perform data access and computation within the same memory operation cycle. This can be achieved using near-memory computing (NMC) or in-memory computing (IMC), as shown in Figs. 2(a) and (b).

1) NEAR-MEMORY COMPUTING (NMC)

In the NMC structure [79], the memory cell array serves as a data storage unit in the same manner as a typical memory device, while a computing circuit block adjacent to the memory cell array performs MAC operations, as shown in Fig. 3(a).

Data stored in the memory cell array are readout by sense amplifiers (SA) to generate a digital output, or by a customized multi-bit readout circuit to generate an analog output for use as inputs to the computing block.

The computing circuit block can be implemented using analog or digital circuits. An external digital input (IN) and weight data (W) read out from the memory cell array are used to perform multiplication, accumulation, and/or place value computation for multi-bit MAC operations. This makes it possible to perform data-access from the memory cell array and MAC computation within the same memory operation cycle, unlike the two cycles required for the conventional von Neumann architecture: (1^{st} cycle) memory access and movement in preparation for the next cycle; and (2^{nd} cycle) MAC computing.

2) IN-MEMORY COMPUTING (IMC)

In the IMC structure [1]–[62], [80], the memory cell array serves not only as a data storage unit, but also performs analog computation, as shown in Fig. 3(b). During MAC computing, each NVM memory cell multiplies a given input by a weight stored in the memory cell. A bitline (BL) then accumulates the multiplication result from memory cells in the same column to generate analog voltage or current. An analog readout circuit, such as a voltage-mode analog-to-digital converter (ADC) or current-mode readout circuit, converts the analog signal on the bitline into a digital MAC output.

Fig. 3(b) illustrates the basic concept and structure of IMC-based nvCIM using single-level-cell (SLC) ReRAM device. In this example, binary inputs are applied to the wordlines (WLs) of multiple accessed memory cells. A WL under high or low voltage is respectively represented by an

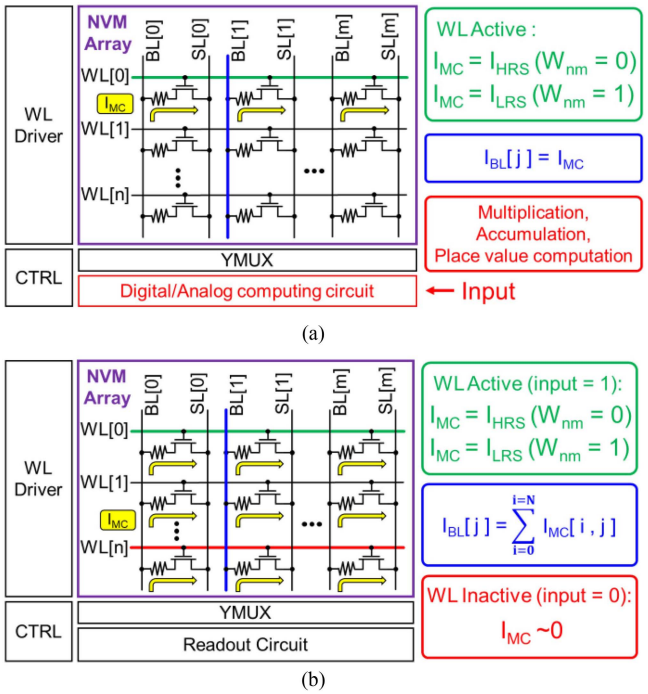

FIGURE 3. Basic concept and structure of (a) NMC and (b) IMC using nonvolatile computing-in-memory (nvCIM).

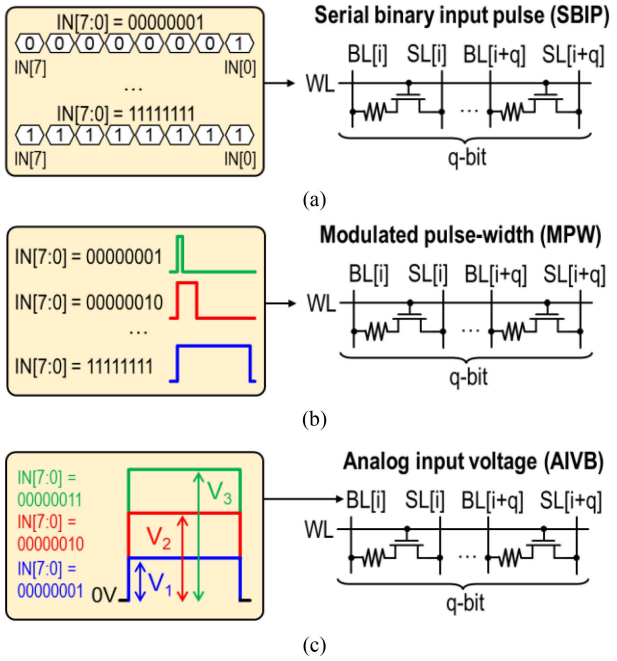

FIGURE 4. Multi-bit input schemes for nvCIM macros: (a) Serial binary input pulse (SBIP); (b) modulated pulse-width (MPW); and (c) analog input voltage via bitline (AIVB).

input value of 1 or 0, which is then multiplied by the weight data (R_{MC}) stored in the ReRAM cell indicating the current of the accessed memory cell (I_{MC}). A multiplication result of "1" indicates that $I_{MC} = I_{LRS}$, whereas a multiplication result of "0" indicates that $I_{MC} = I_{HRS}$ or 0. All current values in a given column are accumulated to generate the current of the accessed bitline (I_{BL}) resulting from the MAC operation. This I_{BL} is then converted into a digital output by a readout circuit.

The IMC scheme is also applicable to multi-level cell (MLC) memory devices, where the 2^N-level resistance state of a given MLC device also serves as N-bit weight data. IMC could potentially provide higher energy efficiency than NMC; however, it poses a number of challenges in terms of circuitry, as discussed in the following subsections.

The remainder of this article focuses on the challenges involved in the design of IMC-based nvCIM as well as its implementation and recent achievements in performance.

C. BASIC APPROACHES TO THE IMPLEMENTATION OF IMC-BASED NVCIM MACROS

1) INPUT METHODOLOGIES

In recent silicon-verified nvCIM devices, multi-bit MAC computation inevitably involves a tradeoff between input bit precision, computing latency, power consumption, and area overhead. NN models that used for testing the nvCIM macros commonly used ReLU activation function, where the inputs are positive or 0. Thus, in the following, we outline the operations and tradeoffs only for positive multibit input methodologies for 1T1R-based NVM devices.

1) Serial binary input pulse (SBIP): The SBIP input scheme [18]–[20], [35] splits *n*-bit inputs into *n* single-bit inputs to be applied serially to the same memory cell, in a process referred to as bitwise MAC computation (see Fig. 4(a)). The SBIP input scheme requires only a simple input driver, and the constraints imposed by process variation are relaxed; however, it also requires *n* serial input pulses for an *n*-bit input. The multi-cycle nature of this input scheme greatly increases the latency of the nvCIM macro in cases where input precision is high.

Take [19] as an example. Each cycle comprises three sub-phases, including control activation (T_{ACT}), analog MAC signal developing (T_{MAC}), and analog readout operation to obtain the digital output (T_{OUT}). In [19], computing latency in one cycle (1-bit input; 11.75ns) breaks down as follows: T_{ACT} (3ns), T_{MAC} (4.5ns), and T_{OUT} (4.25ns). For N-bit input precision, the use of SBIP in [19] imposed latency of N x (T_{ACT}+T_{MAC}+T_{OUT}) for computation.

2) Modulated pulse-width (MPW): The MPW input scheme [1] modulates the pulse-width of a wordline (WL) according to the values of the multi-bit inputs, as shown in Fig. 4(b). The modulated input pulse-width is applied to all selected memory cells to determine the memory cell current (I_{MC}) used in that single computing cycle. The accumulated charge ($Q = I_{MC}xT_{MAC}$) on the activated bitline contributed by a single memory cell is equal to the product of the memory cell current (I_{MC}) and WL pulse-width for MAC computing (T_{MAC}). Thus, MACV refers to the charge accumulated from all of the activated memory cells on a bitline.

In [1], total computing latency in one cycle (14.9ns) breaks down as follows: T_{ACT} (1.5ns), T_{MAC} (0.2ns), and T_{OUT}

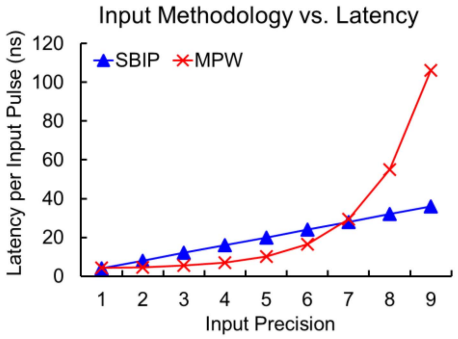

FIGURE 5. SBIP and MPW input method versus input pulse latency.

(2.3ns). The computing latency of MPW (T_{MPW}) for an N-bit input is $T_{ACT}+2^{N}xT_{MAC}+T_{OUT}$. Under the same conditions, the computing latency of SBIP (T_{SBIP}) for an N-bit input is N x ($T_{ACT}+T_{MAC}+T_{OUT}$). When N increases, the term $2^{N}xT_{MAC}$ in T_{MPW} increases exponentially and eventually dominates the overall MAC computing latency for a given cycle. By contrast, the growth of T_{SBIP} is linear (i.e., well-suited to multibit input precision).

Note that the wordline pulse-width must be precisely controlled in accordance with PVT conditions and process variations to prevent a loss of signal margin in readout circuits or even miscomputation. Note that the signal margin mentioned in subsequent sections refers to the minimum difference in voltage or current between two consecutive MAC values (MACVs). As with SBIP, computing latency increases with the input-bit precision. Furthermore, the longest input pulse determines the overall MAC computing latency in that cycle.

Fig. 5 presents the growth in latency for N-bit input precision under the SBIP and MPW schemes. The T_{ACT}, T_{MAC}, and T_{OUT} has normalized to the same value using the given value from [1]. Note that there is a crossover point at 7-bit input precision, which indicates that T_{MAC} has a more pronounced impact on computing latency when input precision is increased.

3) Analog input voltage via bitline (AIVB): AIVB input schemes [2]–[34] use multiple analog voltages for bitline clamping or as the initial voltage (V_{BL}) representing a multi-bit input, as shown in Fig. 4(c). Multi-level bitline voltages alter I_{MC} values in accordance with the V_{BL} and cell-resistance of the accessed memory cell. Note that I_{MC} represents the multiplication result of the input (V_{BL}) and the weight (R_{MC}). Note also that analog input voltage must be below the NVM read-disturb-free voltage to avoid disturbing the memory cell. Thus, MACV refers to the bitline current equal to the accumulated I_{MC} of all activated memory cells.

The AIVB input scheme allows the insertion of multi-bit inputs into a memory array for computation within a single cycle, thereby reducing computing latency to below that of SBIP and MPW, particularly in cases where input precision is high to deal with complex datasets. However, variations in input voltage can lead to fluctuations in I_{MC} during multiplication, and enlarge variations in I_{BL} through

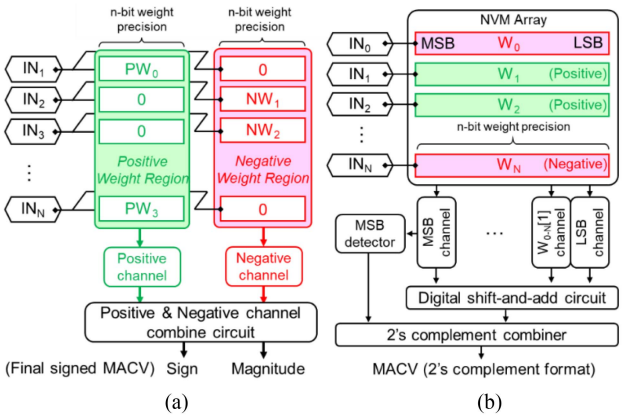

FIGURE 6. Two types of weight data commonly used in nvCIM: (a) Positive and negative weight data; (b) Two's complement weight data.

the accumulation of multiple offsets generated by I_{MC}. This can in turn limit the number of input bits in a single MAC computing cycle. AIVB schemes also require a digital to analog converter (DAC) to generate input voltage, which increases area overhead.

2) FORMATION AND ASSIGNMENT OF WEIGHT DATA

The weight data in typical CNN computing includes both positive and negative values. In the following, we describe two methods used in the assignment of weight data in memory cell arrays, as well as the tradeoffs for nvCIM macros using SLC devices.

1) Positive and negative weight group (PNWG): The PNWG [19], [33], [46] weight storage method respectively assigns positive and negative multibit weights to two groups to enable storage in different banks within the nvCIM macro, as shown in Fig. 6(a). The identical multibit input is simultaneously applied to the positive weight region and negative weight region to perform partial MAC computation using unsigned data. After the pMACV is respectively generated from the positive and negative channels, a positive/negative channel combiner (analog or digital) then combines the unsigned partial MACV values into a final signed MACV using signed-magnitude type data.

To ensure consistency in input data (wordline and row address), identical memory capacity is used for the positive and negative groups. In other words, 2N memory capacity is required to store a N-bit weight data. As a result, PNWG schemes are prone to large area overhead and low memory usage efficiency.

2) Two's complement weight (2's CW): A number of recent silicon-verified nvCIM macros have used the 2's CW [1]–[2], [34] data format for the storage of weight data without separating positive and negative weight data. For an *n*-bit weight, the MSB represents the signed bit and the remaining (n−1) bits represent the unsigned value. These data are stored across *n* memory cells in the same row of a single memory cell array, as shown in Fig. 6(b). The readout circuit reads out the magnitude of the partial MACV (pMAC) computation

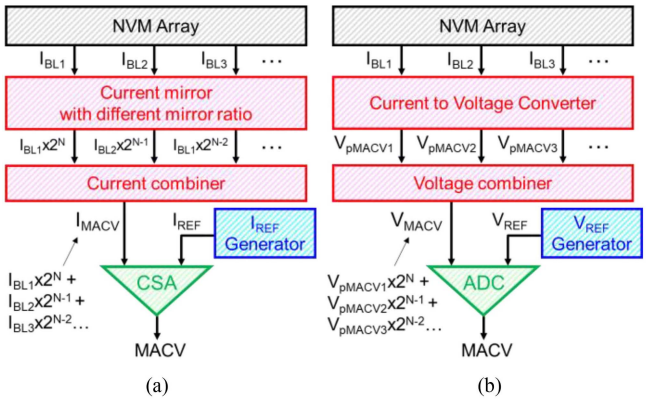

FIGURE 7. Two readout schemes for nvCIM: (a) Current mode readout; (b) Voltage mode readout.

result from each place-value channel. A digital shift-and-add circuit then combines the pMACV from each channel in accordance with its place value. Note that an MSB detector is required to generate the sign bit in the MSB channel. The 2's complement combiner generates the final MACV in 2's complement format in accordance with the outputs from the digital shift-and-add circuit and MSB detector.

3) READOUT AND QUANTIZATION APPROACHES

Converting the analog MACV signal generated on a bit-line into a digital output requires an nvCIM macro with a voltage-mode analog to digital converter (ADC) or current-mode sense amplifier (CSA). Many nvCIM macros suppress the power consumption of the MACV readout by reducing precision using various quantization schemes. In this section, we examine the readout and quantization methods reported in recent silicon-verified nvCIM macros.

1) Current mode readout: As described earlier, the analog MACV in most nvCIM macros is indicated by the bitline current (I_{BL}). The current-mode readout scheme in Fig. 7(a) combines multiple I_{BL} from different columns using current-mirror circuits for place value computation, and then generates a signal current indicating the partial or full MACV. A current mode sense amplifier (CSA) with multiple reference currents (I_{REF}) is then used to convert the current into a digital output of MACV.

The difference in current between neighboring MACV (ΔI_{MACV}) is independent of the value of the power supply voltage (VDD). However, the current-mode readout method tends to increase the energy consumption by imposing multiple DC-current branches in the memory array, current-mirror circuits, and reference-current generator.

2) Voltage mode readout: Fig. 7(b) illustrates the structure of a typical voltage-mode readout module, comprising a current to voltage converter (IV-converter), voltage combiner, voltage-mode ADC, and reference-voltage (V_{REF}) generator. The IV-converter converts each MACV current (I_{BL}) into an analog voltage V_{pMACV}, and then the voltage combiner combines multiple V_{pMACV} values using place values with various input and weight bits to generate analog voltage

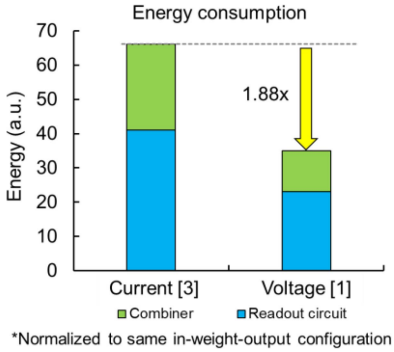

FIGURE 8. Energy consumption of current-mode and voltage-mode combiners (combining analog pMAC signals with different place-values) with a readout circuit.

V_{MACV} representing the partial or full MACV. A voltage-mode ADC then converts the V_{MACV} into a digital output. Voltage-mode readout schemes generally consumes less energy than current-mode readout schemes by eliminating many of the DC-current branches and reducing the signal setting time. However, voltage-mode readout can compromise the signal margin due to the fact that the difference in voltage between neighboring MACV (ΔV_{MACV}) depends on the value of the supply voltage (VDD). Any increase in the precision of MACV leads to a corresponding reduction in ΔV_{MACV}. Furthermore, most voltage mode sense amplifiers do not support readout functionality when the input voltage falls below the threshold voltage of its input transistor (V_{TH-IN}), which further limits the voltage range of analog MAC values.

In implementing IMC-based nvCIM macros, developers have various design choices pertaining to input methodology and readout combiner. In [19], SBIP and a current combiner were used to generate current for MAC operations involving 9 accumulations of 2b-input and 3b-weight. In [3], AIVB and a current combiner were used to generate current for the multiplication of 2b-inputs and 4b-weights. In [1], MPW and a voltage combiner were used to generate analog voltage for pMAC operations involving 4 accumulations of 6b-input and 1b-weight.

Fig. 8 compares the energy consumption of current-mode and voltage-mode combined using the schemes in [1] and [3] as examples. Here, the energy consumption of the voltage-mode combiner and readout circuit was 1.88x lower than that of the current-mode combiner, due to the elimination of DC-current during MAC computing and readout in the analog domain.

Fig. 9 compares the area of the current-mode and voltage-mode readout circuits using the sequential readout methodology for N-bit output. Most sequential current-mode readout circuits include a current-mode sense amplifier (CSA) and current-mode reference generator (CREF-GEN) with 2^{N-1} current branches. The number of VSA and VREF-GEN used in the sequential voltage-mode readout scheme is equal to the number used in current-mode readout scheme. The CSA and CREF-GEN are generally implemented using large transistors to tolerate large fluctuations in input current (I_{BL}) across a wide range of MAC values (MACVs), which results in high area

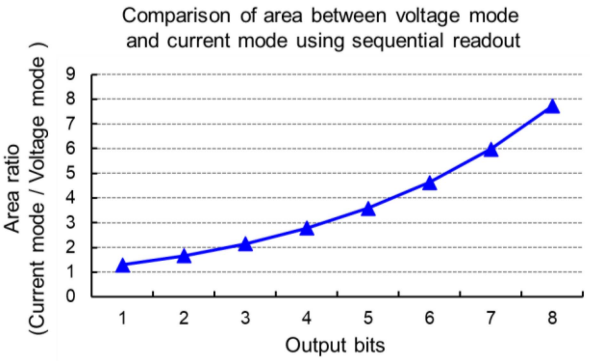

FIGURE 9. Comparison of area between voltage mode and current mode using sequential readout.

overhead. Increasing readout precision leads to a corresponding increase in the area required for the reference branches used in current-mode and voltage-mode readout schemes.

3) Oscillator-based readout: In [80], a current controlled oscillator-based ADC was proposed to digitize current on the BL (representing MACV) by converting the current amplitude into an oscillation frequency. The period of the oscillation is then readout using a time-to-digital converter (TDC) to generate TDC codes, which are then mapped to a corresponding MACV via a digital circuit. Their current-controlled oscillator-based ADC achieved compact area, high speed, and high energy efficiency; however, that approach is still limited by the maximum oscillation frequency and the precision of the time-to-digital converter.

4) Quantization methods for nvCIM: Recent studies have adopted two approaches to quantization: linear and clipped.

Linear quantization [1]–[2], [34] equally divides the original M-value multilevel signal into a N-value multilevel signal, with equal loss of information across all digital readout values, as shown in Fig. 10(a). For example, every two MACVs are combined in order to change a 4-bit 16-level partial-MACV signal into a 3-bit 8-level partial-MACV signal, resulting in the loss of 1 bit.

Clipped quantization [21], [43], [45] merges all of the analog MACVs exceeding a selected threshold (MAC_{TH}) into a single digital value, while the other analog MACVs below MAC_{TH} are read out individually without any quantization loss, as shown in Fig. 10(b). Clipped quantization is suitable for CNN models in which most of the partial MACVs are relatively small, resulting from input data sparsity. However, clipped quantization requires a high-resolution ADC to readout the low MACVs without a loss of precision.

Fig. 11 presents an example pMACV distribution of a scenario involving 16 accumulations of 1b-input, 1b-weight, and 3b-output (reduce 1b) in the inter-channel direction using the ResNet-20 model trained using 8b-input and 8b-weight (2's complement format) for the CIFAR-100 dataset. In general, there is a tradeoff between the precision of pMACV and energy consumption.

In many neural network models (e.g., ResNet-20), the probability of ultra-large pMACV is very low. Thus, the

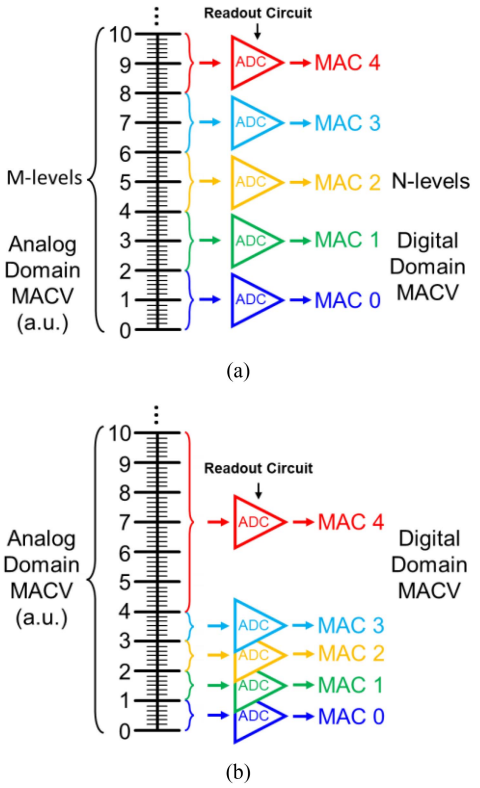

FIGURE 10. Two quantization methods commonly used for nvCIM; (a) Linear quantization; and (b) Clipped quantization.

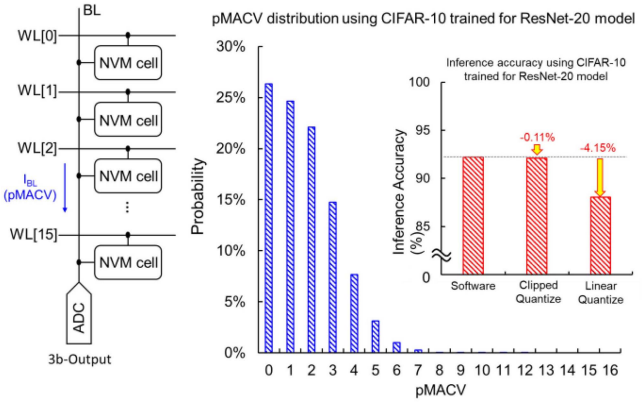

FIGURE 11. pMACV distribution and inference accuracy using the ResNet-20 model trained for CIFAR-10 with various quantization methods.

clamped-quantization method is meant to preserve high-probability pMACVs, while limiting the occurrence of ultra-large pMACVs to below a given threshold in order to reduce energy consumption with only a minor reduction in output precision and an insignificant loss of accuracy. Compared to the pure software approach (without clamped quantization), the clamped-quantization approach was shown to reduce inference accuracy by only 0.11% in tests applying ResNet-20 to the CIFAR-10 dataset. By contrast, the linear quantization approach imposed a 4.15% decrease in inference accuracy for the same reduction in pMACV read precision.

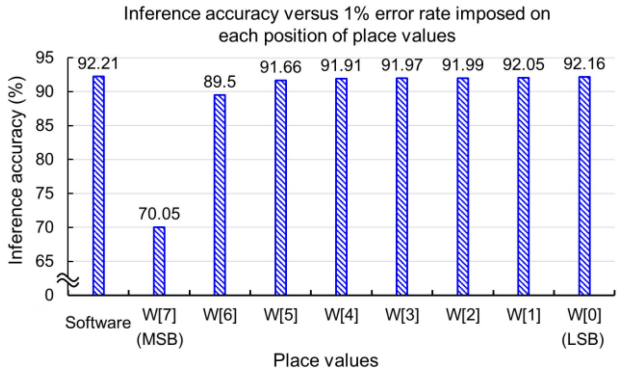

FIGURE 12. Inference accuracy with 1% readout error imposed on various place-value output channels.

The readout error also affects the inference accuracy significantly. When using the weight bit-slicing approach (storing N-bit weights in N SLC devices), errors in the place-value of weight data have a direct effect on inference accuracy; however, the effect tends to vary. For instance, error in the MSB can have a more pronounced impact than does error in the LSB, particularly after bit-shifting in the digital shift-and-add circuit to combine pMACVs with respective place-values. In other words, readout error in the MSB is amplified by the bit-shift operation.

Fig. 12 presents the inference accuracy of a ResNet-20 model trained for the CIFAR-10 dataset using 8b weight with a 1% error rate across various place-values of weight data. Here, 1% error in the MSB of weight data resulted in a 22.16% decrease in inference accuracy after bit-shifting, whereas the same 1% error in the LSB resulted in only a 0.05% decrease in inference accuracy. Clearly, maintaining readout accuracy in the MSB channel is crucial to inference accuracy when using the weight bit-slicing approach [1].

III. DESIGN CHALLENGES FOR IMC-BASED NVCIM

Creating nvCIM macros with high-precision input, weight, and output imposes a number of hinderances, including readout circuits with large area overhead and high energy consumption, small signal margin, limitations on the readout circuit pitch, and a trade-off in throughput. Those challenges are discussed in the following sub-sections.

A. AREA OVERHEAD IN READOUT CIRCUITS

Fig. 13 presents an illustrative example showing the area breakdown of a silicon verified 2Mb ReRAM nvCIM macro [3] employing a current-mode readout scheme for 4b MAC computing. This typical memory circuit with an NVM array, WL drivers, and column multiplexors (Y-MUX) consumed 73.23% of the total macro area. The peripheral computing circuit with analog circuits for MAC computation and a digital shift-and-add circuit for place value computation consumed 20.94%. Under the same Y-MUX scheme, the inclusion of computing-related circuits increased the size of the nvCIM macro to 23.81% larger than that of a conventional ReRAM macro tasked only with storage operations.

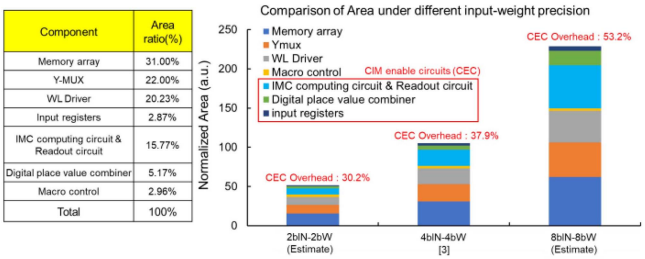

FIGURE 13. Illustrative example of area breakdown in recent silicon-verified IMC-based 2Mb ReRAM nvCIM macro [3].

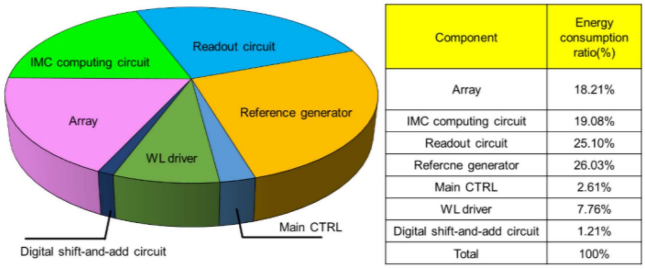

FIGURE 14. Illustrative example showing energy breakdown in a recent silicon verified IMC based 2Mb ReRAM nvCIM macro [3].

Area overhead is dominated by the analog-based computing circuit, readout circuit, and digital place-value combiner. When dealing with CIM, circuit area overhead is generally proportional to the input-weight-output precision. Increasing input precision requires more input registers and DACs to support analog MAC computing. Increasing weight precision tends to increase the amount of memory required for the weight data used in the inference stage. Increasing output precision tends to increase the number of transistors required for computing circuitry and the number of sense amplifiers and reference branches used to generate a multibit digital output. Increased output precision also requires a large-scale digital place value combiner and a larger number of output registers. Area overhead can be reduced by shrinking the number of IOs (deeper Y-MUX); however, this limits computing throughput.

B. ENERGY CONSUMPTION BY READOUT CIRCUIT AND MEMORY CELL-ARRAY

Fig. 14 presents an illustrative example showing the energy breakdown in a 2Mb ReRAM nvCIM macro for 4b MAC computation [3]. Here, energy consumption during MAC operations was dominated by the IMC computing circuit and readout circuit, which account for 71.51% of the total macro energy. The typical memory circuit including the memory array, WL driver, and YMUX consumed 25.88% of the total macro energy.

The energy consumption of a typical memory circuit is dominated by the memory cell array, due to the need to accumulate of memory cell current for MAC computation. Note that the energy consumption of a memory cell array increases with the number of low resistance state (LRS) cells that are accessed.

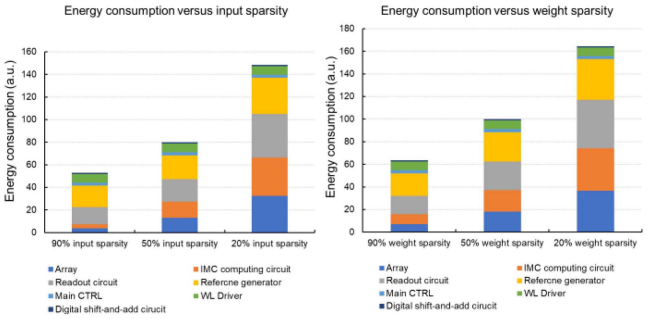

FIGURE 15. **(a) Energy consumption versus input sparsity (90%, 50%, and 20%); (b) Energy consumption versus weight sparsity (90%, 50%, and 20%).**

Energy consumption in the readout circuit is dominated by the DC current path in current-mirror circuits and analog-to-digital conversion (via an ADC or current-mode sense amplifiers) as well as the generation of reference current or voltage signals for multibit readout. Moreover, achieving high readout accuracy when the signal margin is small requires large transistors for the analog readout circuit as well as additional offset suppression circuits to suppress the input offset, both of which increase power consumption.

Achieving high output precision when the signal margin is small requires that the analog readout circuit run through a larger number of operational phases using a larger number of reference voltages or currents, which increases readout latency and power consumption.

Figs. 15 (a) and (b) shows the energy consumption versus input sparsity (90%, 50%, and 20%) and weight sparsity (90%, 50%, and 20%) using [3] as example. Overall, sparsity was inversely proportional to energy consumption in the nvCIM macro. The effect of input sparsity on energy consumption was more pronounced than that of weight sparsity

C. CONSTRAINTS ON SIGNAL MARGIN DUE TO CELL-RESISTANCE VARIATION, READ DISTURB VOLTAGE, R-RATIO, AND DATA-PATTERNS

Emerging NVM devices (particularly MLC NVM devices) vary considerably in terms of cell-resistance (R_{MC}) [74], due to process variation, temperature, and resistance drift. Most of these devices also require low read voltage (V_{read}) on the bitline to prevent data corruption during read operations (i.e., read disturbance). These factors greatly limit the signal margin between neighboring computing states (MACV) and limits the number of operations that can be performed by NVM devices.

Most production-ready NVM technologies also suffer from a limited cell-resistance ratio (R-ratio); i.e., cell-resistance (R_{MC}) in high-resistance state (HRS; $R_{MC} = R_{HRS}$) versus low-resistance state (LRS; $R_{MC} = R_{LRS}$). Under a small R-ratio, the difference in memory cell current between LRS (I_{LRS}) and HRS (I_{HRS}) is small and I_{HRS} is non-negligible.

Non-negligible I_{HRS} can render the bitline current (I_{BL}) of a MACV susceptible to fluctuations in input-weight patterns and the number (N_{WL}) of activated wordline signals.

MACV	NWL	HRS	LRS	I_{BL}
0	0	0	0	0LRS+0HRS
	1	1	0	0LRS+1HRS
	2	2	0	0LRS+2HRS
	3	3	0	0LRS+3HRS
	4	4	0	0LRS+4HRS
	5	5	0	0LRS+5HRS
	6	6	0	0LRS+6HRS
	7	7	0	0LRS+7HRS
	8	8	0	0LRS+8HRS
	9	9	0	0LRS+9HRS
		...		
7	7	0	7	7LRS+0HRS
7	8	1	7	7LRS+1HRS
7	9	2	7	7LRS+2HRS
8	8	0	8	8LRS+0HRS
8	9	1	8	8LRS+1HRS
9	9	0	9	9LRS+0HRS

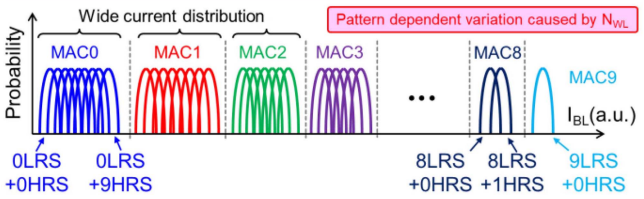

FIGURE 16. **Wide MAC current distribution caused by pattern dependent variation.**

Pattern-dependent variations in bitline current can widen the signal distribution of MACV leading to a decrease in signal margin, as shown in Fig. 16. Consider an example of an nvCIM macro with 9 accumulations per MAC operation. A situation where the I_{BL} for MACV = 2 could be associated with two WLs switched on, involving two LRS and zero HRS cells (designated as 2L0H). Likewise, a situation where the I_{BL} for MACV = 2 could be associated with nine WLs switched on, involving two LRS cells and seven HRS cells (designated as 2L7H). A similar situation where the I_{BL} for MACV = 6 (i.e., 6 accessed LRS cells) could be associated with six WLs switched on, involving six LRS cells and zero HRS cell (6L0H). It could also be associated with nine WLs switched on, involving six LRS and three HRS cells (6L3H). Thus, variations in bitline current are dominated by pattern dependent variations in cases of small MACV, and by cell-resistance variations in cases of high MACV. In cases involving mid-level MACV values, variations in bitline current are contributed by cell-resistance variations as well as pattern-dependent variations.

Fig. 17 presents simulation results of normalized signal margin versus output precision in the analog domain under various R-ratio values. When output precision is increased, pattern dependent variation can significantly decrease the signal margin; however, a large R-ratio can be used to suppress pattern dependent variation. It is also possible to increase the R-ratio by over-setting or over-resetting the NVM cells while programming the device. Moreover, an HRS cancellation circuit can be installed to remove HRS cell current to increase the number of accumulations on a given BL in order to improve the readout yield [3].

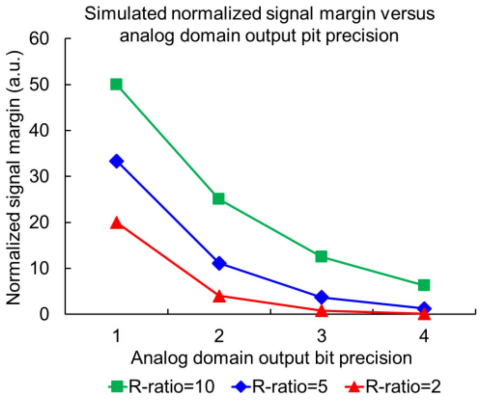

FIGURE 17. Simulated signal margin versus output precision in the analog domain under various R-ratio values.

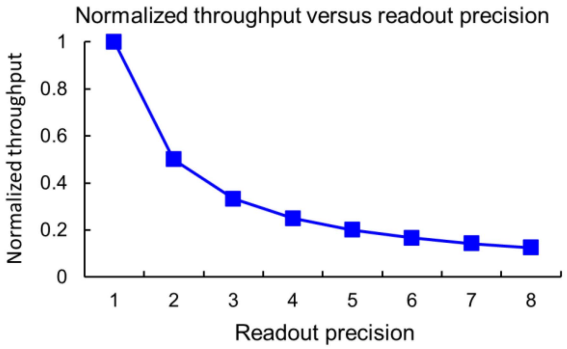

FIGURE 18. Normalized throughput versus readout precision.

D. LIMITATIONS ON READOUT CIRCUIT PITCH AND THROUGHPUT

Typical 1T1R-based NVM cells are compact; however, laying out peripheral circuits in accordance with the required functionality can be hampered by cell-pitch constraints.

Most wordline drivers employing high-voltage devices to support write operations consume greater area than do core-voltage devices. This means that adding computational functions to wordline drivers for input control is more challenging than implementing typical memory applications.

As discussed in the section on area overhead, the area required for the readout circuit tends to increase under higher input, weight and output precision and efforts to suppress layout-dependent device mismatch. When dealing with binary readout precision or a simple readout circuit, the required pitch can be 2:1 [21] or even 1:1 [80]. However, higher precision requires a more complex readout circuit, which increases the area overhead. Due to a small cell-pitch in the column direction, nvCIM macros should use multiple bitlines with column multiplexors (YMUX) sharing the same readout circuits. We can assume that 1-bit readout precision requires 1 pitch for the readout circuit, such that N-bit output requires for N pitches for the readout circuit. Thus, throughput would degrade to 1/N for N-bit readout precision, as shown in Fig. 18.

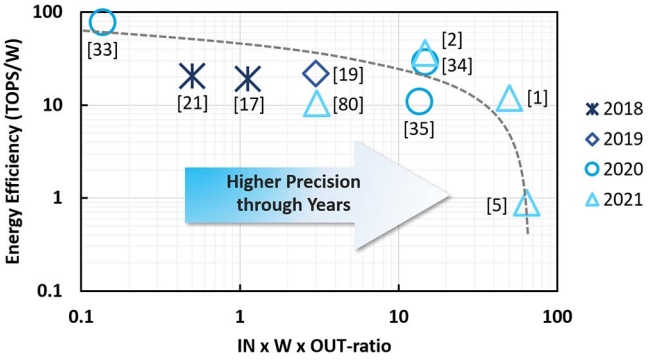

FIGURE 19. Energy efficiency versus product of input precision (IN), weight precision (W), and output-ratio (OUT-ratio).

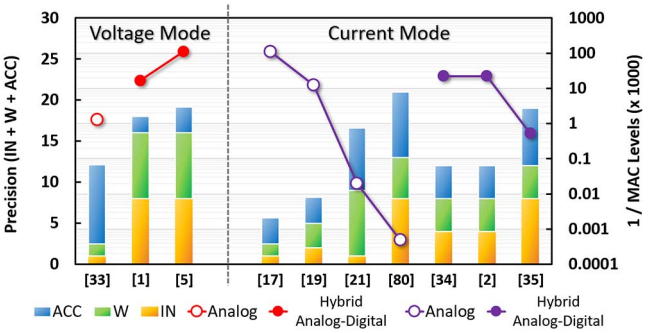

FIGURE 20. Signal margin, derived from the number of MACV levels versus full output precision based on input precision, weight precision, and the binary logarithm of the number of accumulations.

IV. PERFORMANCE OF NVCIM IN MAC COMPUTING

This section examines the performance of recent silicon-verified IMC-based nvCIM macros and the results of design space analysis used to characterize observed trends.

Fig. 19 plots the energy efficiency of MAC operations versus the product (precision-product) of input precision (IN), weight precision (W), and output-ratio (OUT-ratio). Note that the OUT-ratio was calculated from the number of output bits per channel divided by the full readout precision of a MAC operation.

Technological advances tend toward higher inference accuracy to support datasets and neural network models of ever greater complexity; however, a higher precision-product also imposes a penalty in terms of energy efficiency. Note also that for the above-mentioned input schemes, higher precision input also results in a longer execution time. Higher weight precision necessitates a larger number of parallel paths to access memory-cells and bitlines as well as the additional processing of place values for each weight bit and additional analog readout operations. This results in longer computing latency and higher power consumption.

Fig. 20 presents the normalized signal margin of recent silicon-verified IMC-based nvCIM works, including voltage-mode and current-mode readout schemes. The signal margin is represented by the reciprocal of the number of MAC levels that a readout circuit must sense. For voltage-mode readout schemes, the signal margin range is normalized to the supply

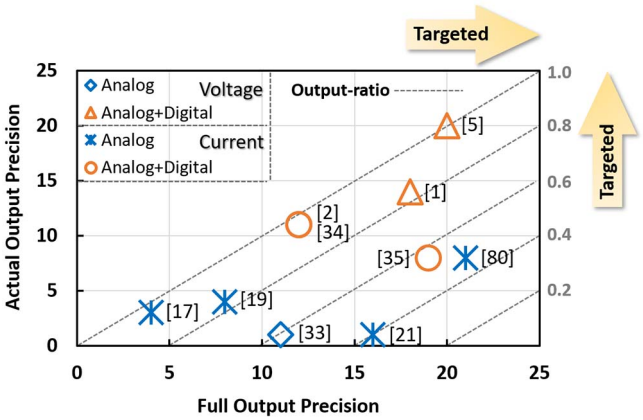

FIGURE 21. Actual output precision versus full output precision where the output-ratio is represented by gray dashed lines.

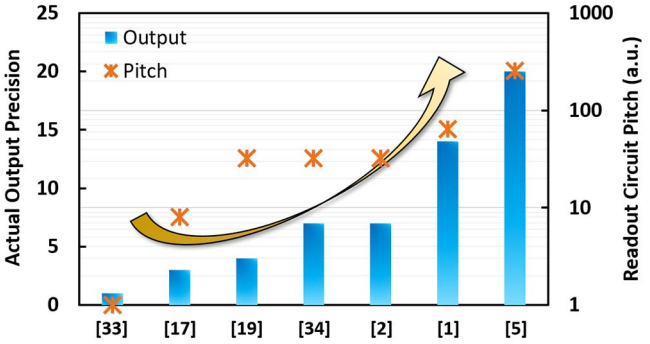

FIGURE 22. Actual-output-precision versus required readout circuit pitch.

voltage. For current-mode readout schemes, the accumulation of MACV DC current results in high power consumption. Thus, it is assumed that the signal margin of current-mode readout schemes is scaled to a fixed range. The stacked bars in Fig. 20 indicate the full output-precision based on input precision, weight precision, and the binary logarithm of the number of accumulations per MAC operation.

In analog CIM schemes, all MAC operations are performed in the analog domain. In hybrid analog-digital CIM schemes, a portion of the MAC operations are performed in the analog domain (with limited bit precision) and the remaining MAC operations are performed by combining partial MAC computing results with digital circuits, such as shifters and adders.

In purely analog CIM schemes, the signal margin is inversely proportional to the full output precision, as shown in Fig. 20. For example, [19] accumulated 9 multiplications of 2b-input and 3b-weight, resulting in signal margin of 2μA without reduced precision. Increasing input and weight precision with a large number of accumulations resulted in an ultra-small signal margin, as indicated by the 256 accumulations of 8b-input and 5b-weight multiplications in [80].

The signal margin of hybrid analog-digital CIM schemes is generally higher than that of purely analog CIM schemes as shown in Fig. 20. For example, the MAC operation in [3] comprises an analog part and a digital part. The analog part involved the multiplication of 4b-input and 4b-weight to output a 7b digital partial MACV. The digital part involved 16 accumulations of digital partial MACV generated from the analog part to output a 11b MACV, resulting in a signal margin of 3μA. The use of hybrid analog-digital readout schemes to enlarge the signal margin and inference accuracy has become an important trend in the development of nvCIM macros

Fig. 21 presents the relationship between full-output-precision (p) and the actual-output-precision (q) of recent nvCIM works based on pure analog and hybrid readout schemes. As mentioned earlier, full-output-precision refers to ideal MACV precision (i.e., without information loss). Actual-output-precision refers the number of output bits implemented by a nvCIM macro, regardless of its full-output-precision. The diagonal line represents the output ratio (p/q).

Due to difficulties in the design of high-resolution readout circuits capable of high output precision, most of the analog-CIM-based nvCIM with high full-output-precision (blue-symbols) failed to achieve a high output-ratio. The nvCIM schemes using hybrid analog-digital CIM (yellow-symbols) achieved a high number of output bits without signal margin degradation during operations in the analog domain.

Fig. 22 illustrates the association between actual-output-precision and the layout pitch of readout circuits in nvCIM macros normalized to the number of typical-size NVM cells. Fig. 22 lists published CIM works using SLC devices. The larger size of non-foundry provided MLC devices used in CIM [21], [35], [80] resulted in a smaller readout circuit pitch, which prevented meaningful comparisons.

nvCIM macros targeting MAC operations with higher input-weight-output precision also require peripheral circuits of greater complexity and larger area overhead. As shown in Fig. 22, an increase in output precision led to a corresponding increase in the layout pitch required for readout circuits. This can be attributed to an increase in the functional complexity of high-bit-precision readout circuits to deal with high-resolution analog readout circuits, small signal margins, and multibit place-value processing. High-resolution analog circuits require multiple large-area transistors or capacitors and a well thought out layout to suppress layout-dependent mismatch in parasitic capacitance and resistance between devices. Thus, the required readout circuit pitch increases as a function of actual-output-precision.

Energy efficiency is a basic benchmark in assessing the performance of nvCIM macros; however, it is not the only judgement criteria. It is important to consider the inference accuracy of AI edge devices in complex applications. As mentioned earlier in Section IV, there is a tradeoff between energy efficiency and MAC precision. Inference accuracy sufficient for most practical applications can only be achieved under relatively high input (IN), weight (W), and output (OUT) bit-precision. We therefore sought to obtain a fair comparison of nvCIM schemes using a figure of

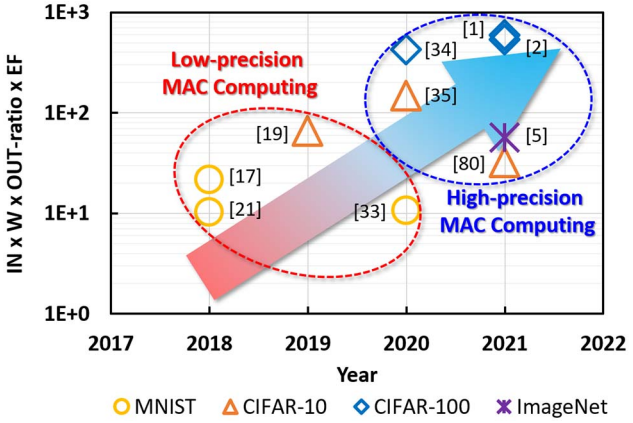

FIGURE 23. Figure-of-the-merit (FoM) of recent published nvCIM works, where the FoM is the product of input precision (IN), weight precision (W), output-ratio (OUT-ratio), and energy efficiency (EF).

merit (FoM) that is the product of input precision (IN), weight precision (W), output-ratio (OUT-ratio), and energy efficiency (EF) (FoM = IN x W x OUT-ratio x EF). As shown in Fig. 23, works prior to 2020 presented high energy efficiency; however, the FoMs were relatively low, due to low input-weight-output precision for simple datasets (e.g., MNIST or CIFAR-10). Despite lower energy efficiency, more recent works achieved higher FoMs due to their higher bit precision, resulting in less data loss when dealing with complex datasets, such as CIFAR-100 and ImageNet. The steady increase in FoM over time provides a realistic indication of the evolution of nvCIM performance.

V. CONCLUSION

nvCIM is a promising candidate to overcome the memory-wall bottleneck and improve the energy efficiency of AI edge devices. This article reviews recent silicon-verified nvCIMs macros based on in-memory-computing in terms of implementation approach and performance as well as on-going challenges pertaining to circuit design. Future nvCIM macros for applications of greater complexity will require greater memory capacity and higher input-weight-output precision; however, this will in turn require novel circuit design techniques to overcome limited signal margin, excessive latency, larger area overhead, and high readout energy.

REFERENCES

[1] C.-X. Xue *et al.*, "16.1 A 22nm 4Mb 8b-precision ReRAM computing-in-memory macro with 11.91 to 195.7TOPS/W for tiny AI edge devices," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2021, pp. 245–247.

[2] C.-X. Xue *et al.*, "A CMOS-integrated compute-in-memory macro based on resistive random-access memory for AI edge devices," *Nat. Electron.*, vol. 4, pp. 81–90, Jan. 2021.

[3] M. Hu *et al.*, "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," in *Proc. 53nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Austin, TX, USA, 2016, pp. 1–6.

[4] Z. Wang *et al.*, "An all-weights-on-chip DNN accelerator in 22nm ULL featuring 24×1 Mb eRRAM," in *Proc. IEEE Symp. VLSI Circuits*, 2020, pp. 1–2.

[5] J.-H. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "29.1 A 40nm 64Kb 56.67TOPS/W read-disturb-tolerant compute-in-memory/digital RRAM macro with active-feedback-based read and *in-situ* write verification," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2021, pp. 404–406.

[6] A. Shafiee *et al.*, "ISAAC: A convolutional neural network accelerator with *in-situ* analog arithmetic in crossbars," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Architect. (ISCA)*, Seoul, South Korea, 2016, pp. 14–26.

[7] S. B. Eryilmaz, D. Kuzum, S. Yu, and H. S. P. Wong, "Device and system level design considerations for analog-non-volatile-memory based neuromorphic architectures," in *IEEE IEDM Tech. Dig.*, Dec. 2015, p. 4.1.1.

[8] S. Yu *et al.*, "Binary neural network with 16 Mb RRAM macro chip for classification and online training," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, San Francisco, CA, USA, 2016, pp. 16.2.1–16.2.4.

[9] M. Bocquet *et al.*, "In-memory and error-immune differential RRAM implementation of binarized deep neural networks," in *IEEE IEDM Tech. Dig.*, 2018, p. 20.6.1.

[10] G. C. Adam, B. D. Hoskins, M. Prezioso, F. Merrikh-Bayat, B. Chakrabarti, and D. B. Strukov, "3-D memristor crossbars for analog and neuromorphic computing applications," *IEEE Trans. Electron Devices*, vol. 64, no. 1, pp. 312–318, Jan. 2017.

[11] M. Prezioso *et al.*, "Modeling and implementation of firing-rate neuromorphic-network classifiers with bilayer Pt/Al2O3/TiO2−x/Pt memristors," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Washington, DC, USA, 2015, pp. 17.4.1–17.4.4.

[12] S. Ambrogio *et al.*, "Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM," *IEEE Trans. Electron Devices*, vol. 63, no. 4, pp. 1508–1515, Apr. 2016.

[13] Y. Liao *et al.*, "Novel in-memory matrix-matrix multiplication with resistive cross-point arrays," in *Proc. IEEE Symp. VLSI Technol.*, Honolulu, HI, USA, 2018, pp. 31–32.

[14] C. Li *et al.*, "In-memory computing with memristor arrays," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2018, pp. 161–164.

[15] C. Li *et al.*, "Large memristor crossbars for analog computing," in *Proc. ISCAS*, Florence, Italy, May 2018, pp. 1–4.

[16] C. Li *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nat. Electron.*, vol. 1, pp. 52–59, Jan. 2018.

[17] W.-H. Chen *et al.*, "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2018, pp. 494–496.

[18] W.-H. Chen *et al.*, "CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors," *Nat. Electron.*, vol. 2, pp. 420–428, Aug. 2019.

[19] C.-X. Xue *et al.*, "A 1Mb multibit ReRAM computing-in-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2019, pp. 388–390.

[20] W.-H. Chen *et al.*, "A 16Mb dual-mode ReRAM macro with sub-14ns computing-in-memory and memory functions enabled by self-write termination scheme," in *IEEE Int. Electron Devices Meeting (IEDM) Dig. Tech. Papers*, 2017, pp. 28.2.1–28.2.4.

[21] R. Mochida *et al.*, "A 4M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture," in *IEEE Symp. VLSI Circuits Dig. Tech. Papers*, 2018, pp. 175–176.

[22] M.-Y. Lin *et al.*, "DL-RSIM: A simulation framework to enable reliable ReRAM-based accelerators for deep learning," in *Proc. ICCAD*, 2018, pp. 1–8.

[23] F. Cai *et al.*, "A fully integrated reprogrammable memristor—CMOS system for efficient multiply–accumulate operations," *Nat. Electron.*, vol. 2, pp. 290–299, Jul. 2019.

[24] B. Chen, F. Cai, J. Zhou, W. Ma, P. Sheridan, and W. D. Lu, "Efficient in-memory computing architecture based on crossbar arrays," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Washington, DC, USA, 2015, pp. 17.5.1–17.5.4.

[25] H. Li *et al.*, "Four-layer 3D vertical RRAM integrated with FinFET as a versatile computing unit for brain-inspired cognitive information processing," in *Proc. IEEE Symp. VLSI Technol.*, Honolulu, HI, USA, Jun. 2016, pp. 1–2.

[26] Z. Wang *et al.*, "Fully memristive neural networks for pattern classification with unsupervised learning," *Nat. Electron.*, vol. 1, pp. 137–145, Feb. 2018.

[27] H. Li *et al.*, "Hyperdimensional computing with 3D VRRAM in-memory kernels: Device-architecture co-design for energy-efficient, error-resilient language recognition," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, 2016, pp. 16.1.1–16.1.4.

[28] Z. Yang and L. Wei, "Logic circuit and memory design for in-memory computing applications using bipolar RRAMs," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Sapporo, Japan, 2019, pp. 1–5.

[29] A. Mohanty, X. Du, P. Chen, J. Seo, S. Yu, and Y. Cao, "Random sparse adaptation for accurate inference with inaccurate multi-level RRAM arrays," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, San Francisco, CA, USA, 2017, pp. 6.3.1–6.3.4.

[30] S. Ambrogio *et al.*, "Novel RRAM-enabled 1T1R synapse capable of low-power STDP via burst-mode communication and realtime unsupervised machine learning," in *Proc. IEEE Symp. VLSI Technol.*, 2016, pp. 1–2.

[31] T. Wu *et al.*, "Brain-inspired computing exploiting carbon nanotube FETs and resistive RAM: Hyperdimensional computing case study," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2018, pp. 492–494.

[32] W. Wan *et al.*, "A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and *in-situ* transposable weights for probabilistic graphical models," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 498–499.

[33] Q. Liu *et al.*, "A fully integrated analog ReRAM based 78.4TOPS/W compute-in-memory chip with fully parallel MAC computing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 500–501.

[34] C.-X. Xue *et al.*, "A 22nm 2Mb ReRAM compute-in-memory macro with 121-28TOPS/W for multibit MAC computing for tiny AI edge devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 244–245.

[35] P. Yao *et al.*, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, pp. 641–646, Jan. 2020.

[36] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, and S. Kvatinsky, "Memristor-based multilayer neural networks with online gradient descent training," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2408–2421, Oct. 2015.

[37] F. Su *et al.*, "A 462GOPs/J RRAM-based nonvolatile intelligent processor for energy harvesting IoE system featuring nonvolatile logics and processing-in-memory," in *Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2017, pp. T260–T261.

[38] C.-C. Chou *et al.*, "A 22nm 96KX144 RRAM macro with a self-tracking reference and a low ripple charge pump to achieve a configurable read window and a wide operating voltage range," in *Proc. IEEE Symp. VLSI Circuits*, 2020, pp. 1–2.

[39] M. Chang *et al.*, "19.4 embedded 1Mb ReRAM in 28nm CMOS with 0.27-to-1V read using swing-sample-and-couple sense amplifier and self-boost-write-termination scheme," in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, 2014, pp. 332–333.

[40] M. Kang, S. K. Gonugondla, S. Lim, and N. R. Shanbhag, "A 19.4-nJ/decision, 364-K decisions/s, in-memory random forest multi-class inference accelerator," *IEEE J. Solid-State Circuits*, vol. 53, no. 7, pp. 2126–2135, Jul. 2018.

[41] W.-S. Khwa *et al.*, "A 65nm 4Kb algorithm-dependent computing-in-memory SRAM unit-macro with 2.3ns and 55.8TOPS/W fully parallel product-sum operation for binary DNN edge processors," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2018, pp. 496–498.

[42] J. Seo *et al.*, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *Proc. IEEE Cust. Integr. Circuits Conf. (CICC)*, San Jose, CA, USA, 2011, pp. 1–4.

[43] J.-W. Su *et al.*, "A 28nm 64Kb inference-training two-way transpose multibit 6T SRAM compute-in-memory macro for AI edge chips," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 240–241.

[44] Q. Dong *et al.*, "A 351TOPS/W and 372.4GOPS compute-in-memory SRAM macro in 7nm FinFET CMOS for machine-learning applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 242–243.

[45] X. Si *et al.*, "A 28nm 64Kb 6T SRAM computing-in-memory macro with 8b MAC operation for AI edge chips," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 246–247.

[46] X. Si *et al.*, "24.5 a twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2019, pp. 396–398.

[47] A. D. Patil, H. Hua, S. Gonugondla, M. Kang, and N. R. Shanbhag, "An MRAM-based deep in-memory architecture for deep neural networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2019, pp. 1–5.

[48] A. F. Vincent *et al.*, "Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 2, pp. 166–174, Apr. 2015.

[49] S. Jain, A. Ranjan, K. Roy, and A. Raghunathan, "Computing in memory with spin-transfer torque magnetic RAM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 3, pp. 470–483, Mar. 2018.

[50] W. Kang, H. Wang, Z. Wang, Y. Zhang, and W. Zhao, "In-memory processing paradigm for bitwise logic operations in STT–MRAM," *IEEE Trans. Magn.*, vol. 53, no. 11, pp. 1–4, Nov. 2017.

[51] H. Cai, Y. Wang, L. A. De Barros Naviner, and W. Zhao, "Robust ultralow power non-volatile logic-in-memory circuits in FD-SOI technology," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 4, pp. 847–857, Apr. 2017.

[52] P. Junsangsri, J. Han, and F. Lombardi, "Logic-in-memory with a nonvolatile programmable metallization cell," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 2, pp. 521–529, Feb. 2016.

[53] M. Natsui *et al.*, "Nonvolatile logic-in-memory LSI using cycle-based power gating and its application to motion-vector prediction," *IEEE J. Solid-State Circuits*, vol. 50, no. 2, pp. 476–489, Feb. 2015.

[54] G. Burr *et al.*, "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element," in *Proc. IEEE Int. Electron Devices Meeting*, San Francisco, CA, USA, 2014, pp. 29.5.1–29.5.4.

[55] S. Kim *et al.*, "NVM neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous *in-situ* learning," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Washington, DC, USA, 2015, pp. 17.1.1–17.1.4.

[56] H. Tsai *et al.*, "Inference of long-short term memory networks at software-equivalent accuracy using 2.5M analog phase change memory devices," in *Proc. Symp. VLSI Technol.*, Kyoto, Japan, 2019, pp. T82–T83.

[57] M. Le Gallo *et al.*, "Mixed-precision in-memory computing," *Nat. Electron.*, vol. 1, pp. 246–253, Apr. 2018.

[58] I. Giannopoulos *et al.*, "8-bit precision in-memory multiplication with projected phase-change memory," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Dec. 2018, pp. 2–4.

[59] C. Mackin, H. Tsai, S. Ambrogio, P. Narayanan, A. Chen, and G. W. Burr, "Weight programming in DNN analog hardware accelerators in the presence of NVM variability," *Adv. Electron. Mater.*, vol. 5, no. 9, 2019, Art. no. 1900026.

[60] D. Ielmini and H. P. Wong, "In-memory computing with resistive switching devices," *Nat. Electron.*, vol. 1, pp. 333–343, Jun. 2018.

[61] W. Zhang, R. Mazzarello, M. Wuttig, and E. Ma, "Designing crystallization in phase-change materials for universal memory and neuro-inspired computing," *Nat. Rev. Mater.*, vol. 4, pp. 150–168, Jan. 2019.

[62] S. Ambrogio *et al.*, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, p. 60, Jun. 2018.

[63] C.-X. Xue *et al.*, "A 28nm 320Kb TCAM macro with sub-0.8ns search time and 3.5+x improvement in delay-area-energy product using split-controlled single-load 14T cell," in *Proc. IEEE Asia Solid-State Circuits Conf. (ASSCC)*, Nov. 2018, pp. 127–128.

[64] C.-H. Chuang *et al.*, "Designs of emerging memory based non-volatile TCAM for Internet-of-Things (IoT) and big-data processing: A 5T2R universal cell," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2016, pp. 1142–1145.

[65] H.-J. Tsai *et al.*, "Energy-efficient non-volatile TCAM search engine design using priority-decision in memory technology for DPI," in *Proc. Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.

[66] L.-Y. Huang *et al.*, "ReRAM-based 4T2R nonvolatile TCAM with 7x NVM-stress reduction, and 4x improvement in speed-wordlength-capacity for normally-off instant-on filter-based search engines used in big-data processing," in *Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2014, pp. 122–123.
[67] J. Li, R. K. Montoye, M. Ishii, and L. Chang, "1 Mb 0.41 μm^2 2T-2R cell nonvolatile TCAM with two-bit encoding and clocked self-referenced sensing," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 896–907, Apr. 2014.
[68] M. F. Chang *et al.*, "17.5 A 3T1R nonvolatile TCAM using MLC ReRAM with Sub-1ns search time," in *Proc. IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, USA, Feb. 2015, pp. 1–3.
[69] S. Matsunaga *et al.*, "A 3.14 m 4T-2MTJ-cell fully parallel TCAM based on nonvolatile logic-in-memory architecture," in *Proc. Symp. VLSIC*, 2012, pp. 44–45.
[70] C.-C. Lin *et al.*, "A 256b-wordlength ReRAM-based TCAM with 1ns search-time and 14x improvement in word length-energy efficiency-density product using 2.5T1R cell," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2016, pp. 136–137.
[71] C. Dou *et al.*, "Emerging memory based circuits for beyond von Neumann applications: Nonvolatile-logic and computing-in-memory," in *Proc. SSDM*, Sep. 2018, p. 9.
[72] K.-T. Tang *et al.*, "Considerations of integrating computing-in-memory and processing-in-sensor into convolutional neural network accelerators for low-power edge devices," in *Proc. Symp. VLSI Technol.*, 2019, pp. T166–T167.
[73] I. Boybat *et al.*, "Neuromorphic computing with multi-memristive synapses," *Nat. Commun.*, vol. 9, p. 2514, Jun. 2018.
[74] W. H. Chen *et al.*, "Circuit design for beyond von Neumann applications using emerging memory: From nonvolatile logics to neuromorphic computing," in *Proc. ISQED*, 2017, pp. 23–28.
[75] J. Yue *et al.*, "A 65nm 0.39-140.3 TOPS/W 1-12bit unified neural network processor using block-circulant enabled transpose-domain acceleration with 8.1x higher TOPS/mm2 and 6T HBST-TRAM based 2D data reuse architecture," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 138–140.
[76] S. K. Gonugondla, M. Kang, and N. Shanbhag, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2018, pp. 490–492.
[77] J. Seo and M. Seok, "Digital CMOS neuromorphic processor design featuring unsupervised online learning," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, 2015, pp. 49–51.
[78] M. Prezioso *et al.*, "Training and operation of an integrated neuromorphic network based on metal–oxide memristors," *Nature*, vol. 521, pp. 61–64, May 2015.
[79] D. Rossi *et al.*, "4.4 A 1.3TOPS/W @ 32GOPS fully integrated 10-core SoC for IoT end-nodes with 1.7μW cognitive wake-up from MRAM-based state-retentive sleep mode," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2021, pp. 60–62.
[80] R. Khaddam-Aljameh *et al.*, "HERMES core—A 14nm CMOS and PCM-based in-memory compute core using an array of 300ps/LSB linearized CCO-based ADCs and local digital processing," in *Proc. IEEE Symp. VLSI Circuits*, 2021, pp. 1–9.
[81] M. Giordano *et al.*, "CHIMERA: A 0.92 TOPS, 2.2 TOPS/W edge AI accelerator with 2 MByte on-chip foundry resistive RAM for efficient training and inference," in *Proc. IEEE Symp. VLSI Circuits*, 2021, pp. 1–2.

JE-MIN HUNG received the B.S. degree in electrical engineering and computer science from National Tsing Hua University, Hsinchu, Taiwan, in 2019, where he is currently pursuing the Ph.D. degree with the Institute of electrical engineering. His current research interests include computing in memory for emerging non-volatile memory.

CHUAN-JIA JHANG (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2020, where he is currently pursuing the Ph.D. degree with the Institute of electrical engineering. His current research interests include computing-in-memory with static random access memory and emerging nonvolatile memory, and neuromorphic circuit design.

PING-CHUN WU (Graduate Student Member, IEEE) received the B.S. degree in interdisciplinary program of science (physics and electrical engineering) from National Tsing Hua University, Hsinchu, Taiwan, in 2020, where he is currently pursuing the Ph.D. degree with the Institute of electronic engineering. His current research interests include computing-in-memory with static random access memory and emerging nonvolatile memory, and neuromorphic circuit design.

YEN-CHENG CHIU received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2018, where he is currently pursuing the Ph.D. degree with the Institute of Electrical Engineering. His current research interests include circuit design of SRAM, memory security, and emerging nonvolatile memory.

MENG-FAN CHANG (Fellow, IEEE) received the M.S. degree from Pennsylvania State University, USA, and the Ph.D. degree from National Chiao Tung University, Taiwan.

He is currently a Distinguished Professor with National Tsing Hua University and the Director of Corporate Research with TSMC. Prior to 2006, he worked in industry for over ten years. This included the design of memory compilers (Mentor Graphics; from 1996 to 1997) and the design of embedded SRAM and Flash macros (Design Service Division of TSMC; from 1997 to 2001). In 2001, he Co-Founded IPLib in Taiwan, where he developed embedded SRAM and ROM compilers, Flash macros, and Flat-cell ROM products until 2006. His research interests include circuit design for volatile and nonvolatile memory, ultra-low-voltage systems, 3D-memory, circuit-device interactions, spintronic circuits, memristor logics for neuromorphic computing, and computing-in-memory for artificial intelligence. He has been the recipient of several prestigious national-level awards in Taiwan, including the Outstanding Research Award of MOST-Taiwan, the Outstanding Electrical Engineering Professor Award, the Academia Sinica Junior Research Investigator Award, and the Ta-You Wu Memorial Award. He has been serving as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS, and IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. He has also been serving on the Executive Committee for IEDM, as well as the Subcommittee Chairs for ISSCC, IEDM, DAC, ISCAS, VLSI-DAT, and ASP-DAC. He was a Distinguished Lecturer for IEEE Solid-State Circuits Society and Circuits and Systems Society (CASS) as well as the Chair of the Nano-Giga Technical Committee of CASS, and an Administrative Committee Member of the IEEE Nanotechnology Council. He has been serving as the Program Director for the Micro-Electronics Program with the Ministry of Science and Technology in Taiwan as well as the Chair of the IEEE Taipei Section, the Associate Executive Director for Taiwan's National Program of Intelligent Electronics (NPIE), and NPIE Bridge Program.