In-Memory Vector-Matrix Multiplication in Monolithic Complementary Metal–Oxide–Semiconductor-Memristor Integrated Circuits: Design Choices, Challenges, and Perspectives

Amirali Amirsoleimani, F. Alibart, Victor Yon, Jianxiong Xu, M. Reza Pazhouhandeh, Serge Ecoffey, Yann Beilliard, Roman Genov, Dominique A Drouin

In-Memory Vector-Matrix Multiplication in Monolithic CMOS-Memristor Integrated Circuits: Design Choices, Challenges, and Perspectives

Amirali Amirsoleimani[1,*], Fabien Alibart[2,3,4,*], Victor Yon[2,3], Jianxiong Xu[1], M. Reza Pazhouhandeh[1], Serge Ecoffey[2,3], Yann Beilliard[2,3], Roman Genov[1], and Dominique Drouin[2,3]

[1]Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada
[2]Institut Interdisciplinaire dInnovation Technologique (3IT), Universite de Sherbrooke, Sherbrooke, Quebec J1K 0A5, Canada
[3]Laboratoire Nanotechnologies Nanosystemes (LN2) CNRS UMI-3463 3IT, Sherbrooke, Quebec J1K 0A5, Canada
[4]Institute of Electronics, Microelectronics and Nanotechnology (IEMN), Universite de Lille, 59650 Villeneuve dAscq, France
[*]amirali.amirsoleimani@utoronto.ca, fabien.alibart@usherbrooke.ca

ABSTRACT

Mining big data to make predictions or decisions is the main goal of modern artificial intelligence (AI) and machine learning (ML) applications. Vast innovation in algorithms, their software implementations and data management has enabled great progress to date, but wide adoption has been slowed by limited capabilities of existing computing hardware. The low communication bandwidth between memory and processing units in conventional von Neumann machines does not support the requirements of emerging applications that rely extensively on large sets of data. More recent computing paradigms, such as high parallelization and near-memory computing (e.g., in GPUs) help alleviate the data communication bottleneck to some extent, but paradigm-shifting concepts are required. In-memory computing has emerged as a prime candidate to eliminate this bottleneck by co-locating the memory and processing. In this context, resistive switching (RS) memory devices is a key promising choice, due to their unique intrinsic device-level properties enabling both storing and computing with a small, massively-parallel footprint at a low power. Theoretically, this directly translates to a major boost in energy efficiency and computational throughput, but various practical challenges remain. We present a qualitative and quantitative analysis of several key existing challenges in implementing high-capacity, high-volume RS memories for accelerating the most computationally demanding computation in ML inference – that of vector-matrix multiplication (VMM). Monolithic integration of RS memories with CMOS integrated circuits is presented as the core underlying technology. We review key existing design choices in terms of device-level physical implementation, circuit-level design, and system-level considerations, and provide an outlook for future directions.

1 Introduction

The semiconductor technology sector, and particularly its research core, are currently undergoing fundamental changes. After decades of predictable evolution based on the strategy relying on CMOS scaling[1] yielding gradual processor performance improvements, new and novel solutions are required[2]. The first driving force for this revolution is energy consumption, which remains a major challenge for the ubiquitous deployment of electronic chips on an ever-increasing number of devices[3]. Solving this challenge would enable both: the integration of more computing functions on a variety of portable miniaturized devices with demanding energy/form-factor constraints, and more generally, conserving the total energy required to power billions of electronic devices. The second driving force is the massive deployment of artificial intelligence (AI) in our everyday lives, which is redefining the basic principles of the hardware architecture required for computing. In particular, the von Neumann computing architecture[4] is not well adapted to machine learning (ML) implementation, which is a main vector for the widespread adoption of AI. Indeed, implementations of ML algorithms on standard CPUs are typically inefficient in terms of speed due to the constant dataflow between arithmetic units (AUs) and memory, limited by the von Neumann bottleneck. There is, consequently, an important need to improve computing efficiency from both energy consumption perspective as well as throughput perspective. To this end, hardware innovation is expected to play a major role by offering viable solutions to sustain the deployment of electronics.

Specialized hardware such as GPUs[5], which are highly parallelized versions of classical von Neumann CPUs, have been game changers in the acceleration of ML. However, they are offering only a partial solution to the speed and energy challenges. More precisely, GPUs are a first step toward hardware specialization where the key operation of Multiply and Accumulate (MAC) has been parallelized in order to offer important speed improvements. Since MAC operation represents the most intensive calculation required for ML algorithm implementation, it explains why GPUs have led to important breakthroughs in acceleration of ML by enabling training and operation of deep neural networks[6] in a reasonable amount of time. But parallelization alone cannot solve the energy challenge for two reasons: (1) Intensive data movement between the different physical elements of the hardware results in important energy consumption (i.e. data movement between on-chip memory and AU, but also data movement in between the different on-chip and off-chip memory level) [7]; (2) As in CPU, the fundamental algorithmic operation is still realized with the same elementary logical operations, which require the same energy budget.

Improving both energy and speed requires rethinking more deeply hardware design principles in addition to prudently exploring emerging computing technologies. Along this line of inquiry, more advanced solutions exploit hardware specialization even further and propose to design application processing units (APUs), which optimize the throughput and energy requirement for a specific application (Figure 1(a)). In these approaches, innovation is supported more by hardware diversification and specialization, rather than by software innovation to create a balance between their functional flexibility and performance [2]. By deploying hardware specialization, there have been several low power research chips, data center chips and cards proposed in addition to recent advancements in CPUs and GPU-based neural engines. However, it should be noted that reaching an end-to-end solution (E2ES) for an efficient hardware will require scrutinizing other computing paradigms and technologies. In this context, in-memory computing architectures enable efficient computing with negligible data movement by co-locating the memory and processing units. This path has been explored with various technological solutions, from mainstream static random access memory (SRAM) and dynamic random access memory (DRAM) to more emerging ones such as embedded dynamic random access memory (eDRAM)[8]. Beyond charge-based digital memory technologies, in-memory computing based on non-volatile resistive switching (RS) devices monolithically integrated on CMOS is opening new perspectives for ultra-efficient MAC operation engine development [9]. Firstly, monolithic integration of memory in close vicinity of logical units reduces significantly the distance for data trafficking, and thus should reduce energy consumption and throughput limitation[9–11]. Secondly, in-memory computing represents a new physical implementation of the basic MAC operation with potential for important improvements with respect to the same criterion.

In this paper, we will review the main limitations and opportunities of in-memory computing with resistive memories for MAC operation engine, also known as Vector Matrix Multiplication engine (VMM engine). On this basis, as shown in Figure 1(b), the challenges hindering the path of monolithically integrated resistive memory and CMOS VMM engines becoming mainstream computing hardware have been categorized into three different levels: physical constraints; circuit-level challenges; and system-level challenges. Initially, we define the main issues corresponding to physical limitations of this specific class of hardware e.g. accuracy; integration; scalability; and speed. Next, we assessed the circuit-level challenges and analyzed the input and output circuit design costs and opportunities. Finally, system-level obstacles such as data movement and data conversion issue have been discussed. Also, we propose a rational analysis of such APUs performance and their trade-offs in the context of ML applications, but the same reasoning could be applied to a wider range of applications[12] such as image processing[13,14], combinatorial optimization[15–19], sparse coding [20,21], associative memory[22–26], deep learning inference/training[27–30], unclonable functions[31–34], principle component analysis[35,36], spiking neural networks[37–41], solving linear[42], and partial differential equations[43] and reservoir computing[44–46]. Our intent is to provide a comprehensive analysis to assess the novelty of the reviewed examples and to discuss different design choices to better understand this emerging class of hardware and to rationalize performances evaluation.

2 Physical implementation of in-memory computing for VMM with resistive memories

2.1 Background

VMM is the main operation module required to implement a neural network structure (Figure 2(a)). The first basic function required for VMM's physical implementation is the multiplication between two real numbers a and b ($a \times b = c$). In digital logic, multiplication is realized by pipelining multiple full-adders (Figure 2(b)). The precision of the multiplication is defined by the digital representation of the real numbers (number of bits, floating/fixed point). Resistive memory, on the other hand, offers a new concept for implementing multiplication leveraging Ohm's law where the current I is equal to voltage V multiplied by conductance G ($V \times G = I$) (Figre 2(b-c)). The advantages of this approach are two-fold: (1) Only a single time step is required to compute the multiplication versus multiple

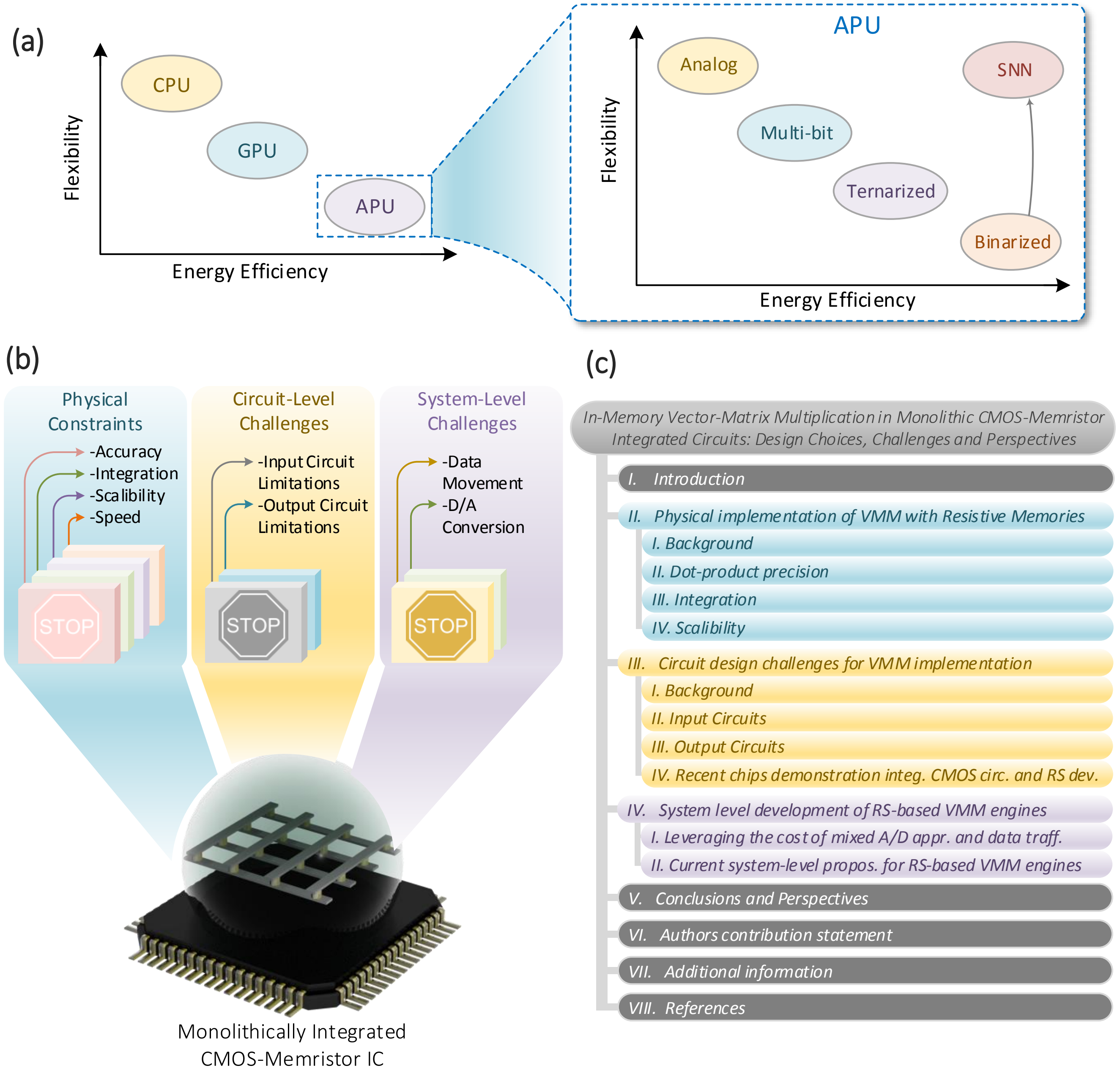
(a)
Flexibility
CPU
GPU
APU
Energy Efficiency
APU
Flexibility
Analog
Multi-bit
Ternarized
SNN
Binarized
Energy Efficiency
(b)
Physical Constraints
-Accuracy
-Integration
-Scalibility
-Speed
STOP
Circuit-Level Challenges
-Input Circuit Limitations
-Output Circuit Limitations
STOP
System-Level Challenges
-Data Movement
-D/A Conversion
STOP
Monolithically Integrated CMOS-Memristor IC
(c)
In-Memory Vector-Matrix Multiplication in Monolithic CMOS-Memristor Integrated Circuits: Design Choices, Challenges and Perspectives
I. Introduction
II. Physical implementation of VMM with Resistive Memories
I. Background
II. Dot-product precision
III. Integration
IV. Scalibility
III. Circuit design challenges for VMM implementation
I. Background
II. Input Circuits
III. Output Circuits
IV. Recent chips demonstration integ. CMOS circ. and RS dev.
IV. System level development of RS-based VMM engines
I. Leveraging the cost of mixed A/D appr. and data traff.
II. Current system-level propos. for RS-based VMM engines
V. Conclusions and Perspectives
VI. Authors contribution statement
VII. Additional information
VIII. References

Figure 1. Various computing hardware performance overview as well as challenges and limitations hindering the path for monolithic CMOS-memristor VMM integrated circuits to become mainstream AI hardware. (a) A simple view of application processing unit (APU) platform's energy efficiency performance and its flexibility in terms of the application versatility is compared with conventional platforms like CPUs and GPUs. Different RS-based APU classes with low to high resolution weight networks are depicted in terms of energy efficiency and application spectrum flexibility. At the opposite of the trade-off between flexibility and energy that existing hardware exhibit, Spiking Neural Networks (SNN), inspired by biology combine both flexibility and low energy consumption. Finding the keys for this implementation may be a disruptive direction for future hardware design. (b) The challenges have been divided into three different categories: physical constraints, circuit-level challenges, and system-level challenges. (c) Manuscript organization.

time steps in digital implementation and (2) Energy consumption is considerably lower. Projected resistive memories performance for an average resistance of $R = 1\text{M}\Omega$ with a read voltage of 0.1V with pulse duration of 1 ns, the energy consumption equals $E_1 = 0.1 \times 10^{-7} \times 10^{-9} = 10^{-17}$J. Note that with today's performances, the energy calculation

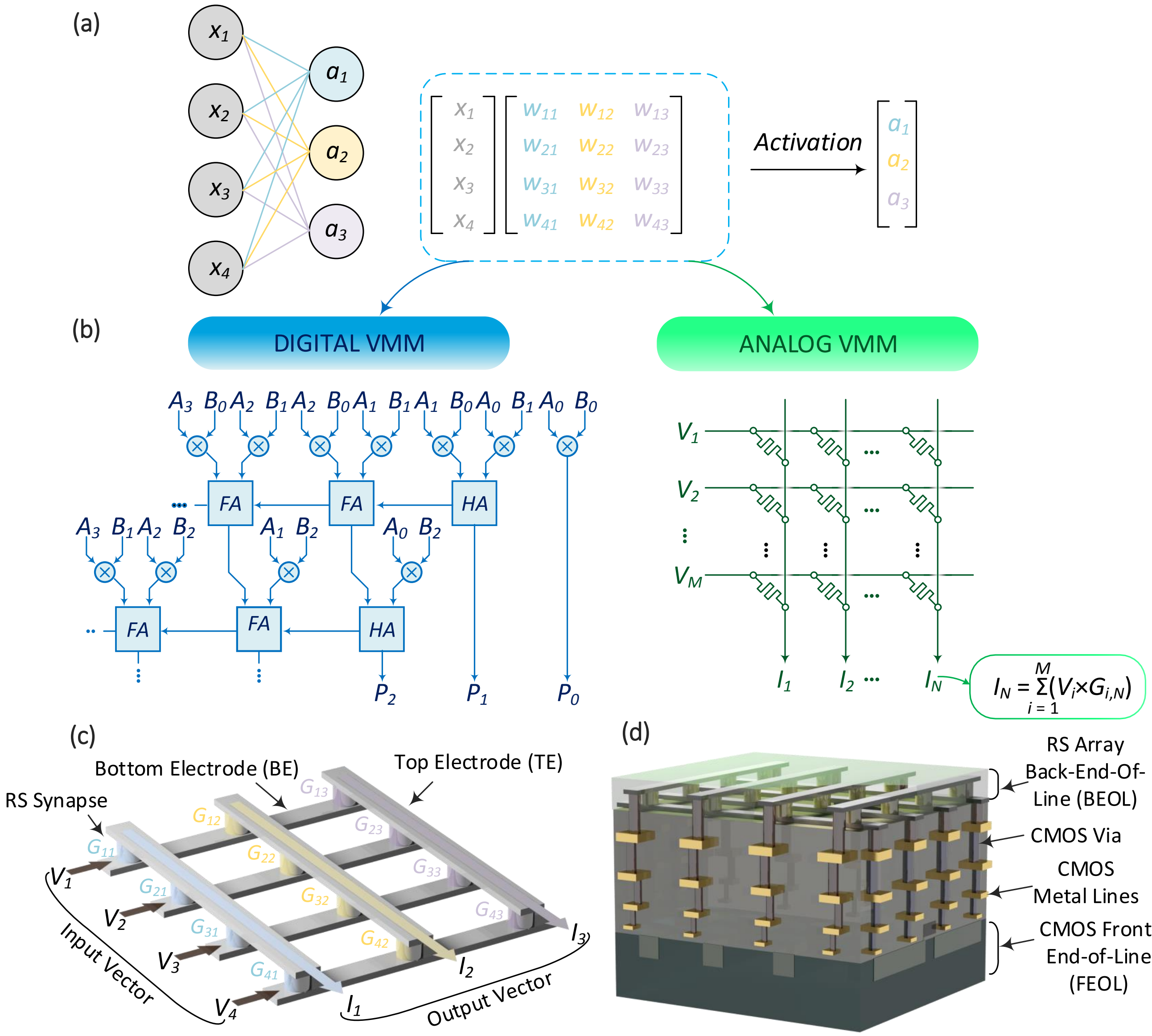

Figure 2. (a) A basic neural network structure is shown including the input vector, weight matrix and output vector. (b) Schematic of the digital and analog vector-matrix multiplication and their implementations. VMM digital implementation is realized by pipelining multiple adders and multiplier digital blocks. Analog VMM on $M \times N$ RS-based crossbar is realized by summing currents from M lines in N columns. (c) The physical implementation of RS-based VMM engine shows the input vector is applied as a voltage vector into the word-lines of the array (bottom electrode (BE)), the weight matrix is stored as the RS devices conductances and the output is sensed as accumulated current in the bit-line (top electrode (TE)). (d) 3D illustration of an RS-based crossbar monolithically integrated on top of the CMOS substrate using a back-end-of-line (BEOL) process.

should consider $R = 10 - 100\text{k}\Omega$, $V = 0.1\text{V}$ and $t = 1\mu\text{s}$ leading to $E_2 = 0.1 \times 10^{-5/-6} \times 10^{-6} = 10^{-12/-13}\text{J}$. This energy consumption should be compared with 8-bit digital multiplication of $E_3 = 0.2\text{pJ}$ with 45 nm CMOS technology node[2] pointing out the important gain attainable only if resistive memory improvement is sustained.

The second basic operation required by VMM is addition. While this operation is carried out by adders in digital electronics, this can also be implemented physically in the analog domain by summing all currents resulting from each multiplicative element in a shared metal line (Kirchhoff's law). This strategy shows a clear advantage for speed improvement due to its highly parallel manner as the Add operations are carried out within multiple parallel channels of the crossbar simultaneously in a single clock cycle with the multiplications. For the sake of comparison, one 8-bit full-adder uses approximately 200 gates in conventional CMOS design and requires a number of computing cycles that are proportional to the Add operation's precision. These two basic multiplication and addition operations correspond to the fundamental MAC operation or dot-product, which constitutes the core of VMM. While this qualitative analysis highlights the advantages in terms of speed and energy consumption of in-memory computing for VMM engine implementation, a fair comparison with digital CMOS technology is more complex and limitations will start to appear due to non-ideal parameters such as physical constraints, overhead circuit design, and system level operation.

2.2 Dot-Product Precision

Resistive switching devices have been developed following two main research directions. On the one hand, resistive switching mechanism has been investigated as a potential solution for the development of a universal memory. This kind of binary memory, called Resistive Random Access Memory (RRAM), could combine high switching speed (sub-ns), low energy (pJ range) and high endurance (1012 cycles) of DRAM and SRAM with non-volatility (>10 years retention) and scalability (<10 nm)(Figure 3(a-b)). Various RRAM cell candidates, among which HfOx and TaOx RRAM are the best representatives (Figure 3(c-d)), are already integrated in fabrication lines of industry and integrated with CMOS technology[47]. They take advantage from CMOS technological maturity and reliability and have been exploited mostly in digital applications such as storage class memories (i.e. Flash). Some recent works have investigated the possibility of storing a few discrete conductance levels in a single memory cell resulting in up to 3-bit multi-level cells. This kind of device can either implement a 1-bit dot-product or a low resolution, e.g. <3-bit dot-product[48].

On the other hand, many research groups have focused on resistive switching mechanism for memristor or memristive device implementation (Figure 3(d)). The association between the theoretical concept proposed by Chua [49] and a possible physical implementation of this new circuit element [50] has opened new perspectives for circuit design, and especially for VMM. In the ideal memristor framework, resistive switching is used to implement a variable resistor where continuous resistive states can be reached by controlling the voltage (or current) applied to (through) the switching material. In that scope, the number of conductance states that can be stored in the memristive element directly defines the precision of the in-memory dot-product computation. In recent years, optimization of the memristive device has focused on the resolution and controllability of the analog switching using various switching mechanisms and materials such as transition metal oxides, ferroelectric tunnel junctions or more exotic materials (See[51] for a review of the different options). Memristive devices have demonstrated analog switching controlled by analog pulses of voltage equivalent to 8-bit accuracy, paving the way for 8-bit dot-product[52]. The 8-bit accuracy has been demonstrated on discrete devices and only 4-bit to 5-bit resolution has been reported for integrated devices due to parasitic effects induced from other circuit elements[53].

The maturity of memristive technologies is not as developed as the RRAM technology, which results in inferior performance regarding endurance, retention, and speed. There are still several research opportunities in this area and efforts need to be pursued to improve memristive devices' overall performance. However, there is currently no strategy nor materials enabling reaching the 32-bit dot-product precision offered by digital approaches. This imposes limitations in terms of VMM applications, such as deep neural networks that rely deeply on the high accuracy calculation of the synaptic weights during training [54]. In that scope, innovations in integration schemes could greatly improve the accuracy of the memristor-based VMM. For instance, while RRAMs differ from analog memristive devices by the difficulty to access to intermediate resistance states, there is, in principle, no physical limitation to have multi-level analog states in RRAM. HfOx-based RRAM, which usually exhibits sharp SET and semi-gradual RESET[55], can be better controlled by using analog current limitation mechanism through an access transistor to implement analog switching close to 5-bit precision[56]. The trade-off here is between a more complex cell design and a higher precision of programming. Along this line, one interesting approach proposed by[57] utilized a hybrid architecture, where two phase change memories (PCM) resistive cells are coupled with six transistors and one capacitor (1C6T2R). Small weight increments, or decrements, are accumulated on a capacitor and stored back in the non-volatile resistive element once accumulated changes fall within the resolution range. Such integration widens the range of VMM applications like in-situ training while decreasing energy consumption compared to contemporary von Neumann architectures. This resolution improvement comes at the cost of more complex resistive cells design and additional shared control circuitry. Short- and mid-term efforts should be dedicated to more complex resistive cells design that would leverage design complexity with controllability and precision for analog VMM implementation.

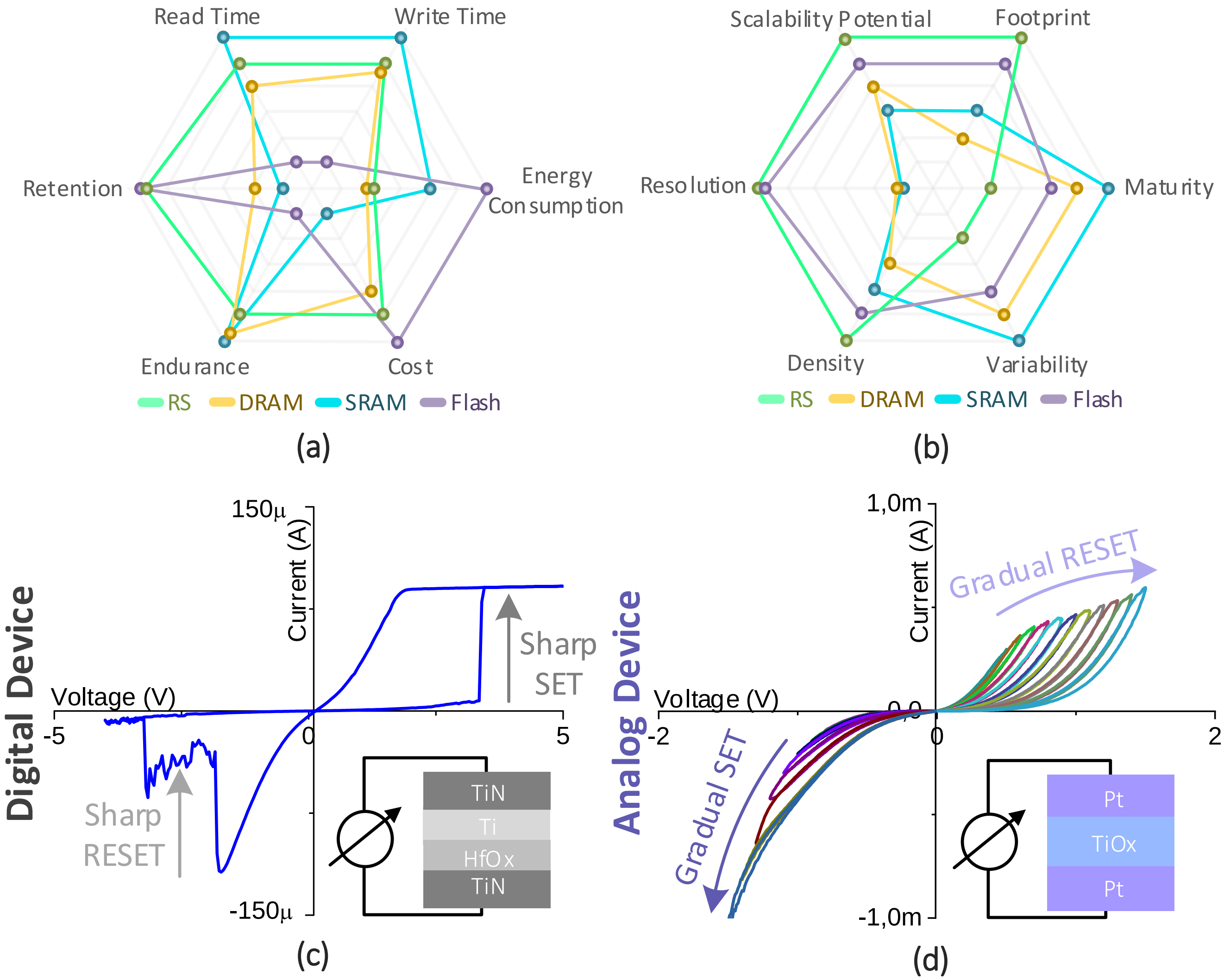

Figure 3. (a) In this spider diagram, resistive switching memories, DRAM, SRAM, and Flash memories are compared in terms of the cost, read time, write time, energy consumption, endurance and retention. (b) In this diagram, the same memories are compared in terms of other criteria: flexibility, footprint size, maturity, density, variability, and potential of the scalability. (c) The $i-v$ curve of the prototypical digital HfOx device with its sharp switching behavior in SET and RESET regions is depicted. (d) The switching behavior for the prototypical memristive TiOx analog device [58] is shown. Both RRAM and memristive devices belong to the resistive switching memories family and described in the spider diagrams in sub-figures (a) and (b) .

2.3 Integration

One of the substantial advantages of RS devices is their advanced integration potential thanks to their excellent scalability. Sub-10 nm switching crosspoints have been reported in[59] and[60], paving the way to surpass the scaling limitations of Flash and DRAM. In addition, the two-terminal structure of RS devices enables ultra-dense integration in crossbar arrays, in which a memory device is located at each intersection between two metallic wires resulting in a matrix-like organization. Finally, RS devices and crossbar arrays can be fabricated with CMOS high-volume manufacturing processes and materials allowing monolithic 3D integration in CMOS BEOL. This ideal approach (see Figure 2(d)) results in a $4F^2$ footprint for a single memory crosspoint, F being the critical dimension of the metal line interconnect). Monolithic 3D BEOL integration of resistive memories presents a major advantage compared to other on-chip memory technologies such as SRAM, which requires a footprint of $200F^2$ in the front-end-of-line (FEOL). This very attractive approach could relax CMOS scaling requirements by providing additional integration opportunities in the vertical dimension. In addition to BEOL attractiveness, the possibility to stack multiple crossbars on top of each other has been demonstrated experimentally and could be conveniently integrated with CMOS for ultra-high-density memory circuit design[33, 61]. There are still important engineering challenges to address in order to bring these concepts to their full potential: (1) Compatibility of advanced lithography steps with BEOL metal layout; (2) Impact of monolithic 3D fabrication processes on the performance of previously fabricated devices; (3) Process homogeneity and yield ensuring high-quality fabrication for each layer; and, (4) High-conductivity interconnects even for ultra-fine pitch. While crossbar architecture offers a truly parallel organization that could map directly the VMM operation, the main limitation comes from the difficulty to access individual memory cells accurately. Parasitic sneak path, currents coming from other resistive cells in the array, are preventing an accurate reading of each resistive element individually. RRAM and memristive devices can be addressed with or without the use of a selector. On the one hand, RRAM requirements have favored optimizations towards accessibility and controllability of an individual memory cell by adding a selector, usually a FEOL transistor, in a series with the two-terminal element leading to 1T1R cells. This solution requires a transistor per memory cell with the allocation of additional silicon area and interconnects for memory management, decreasing the attractiveness of two terminal resistive memory. In addition to this, for fast switching ($<$ 10ns) during the device programming and performing a successful device forming operation, higher voltage amplitudes (3-4 V) in comparison with CMOS regular working voltage (1-2 V) is required. Using a thick-oxide transistor to tolerate this high voltage is another co-integration issue[59] in these platforms and it brings a high area cost. The resulting integration scheme is then only considered as a pseudo-crossbar array. Two-terminal selectors, such as threshold switching elements or non-linear diodes, are currently attracting lots of attention to 1S1R cells. Those passive elements can prevent sneak path currents and preserve the two-terminal interconnection of each memory cell[62]. Still, 1S1R integration is facing important challenges such as (1) Large variability coming from the selector itself; and, (2) Shorter endurance in the case of switching selectors that need to be switched for each read operation. Detailed review in this topic can be found in[63]. On the other hand, memristor-based approaches for physical VMM have favored the concept of selector-less passive crossbar integration. While RAM operations require precise access to an individual memory cell, the memristor-based dot-product is different since this operation is not affected by sneak paths (e.g. all lines and columns are polarized at the same time and all resistive cells are read at the same time). More exploratory in-memory computing paradigms such as neuromorphic computing, or bio-inspired spiking neural networks, can also take advantage of a similar principle. The trade-off is, therefore, to favor parallelism and aggressive integration at the cost of less accurate access to individual crosspoints sequentially. It should be noted that practical integration of crossbar on chip still requires access transistors at the N input lines and M output columns of the crossbar thus lead to $(N+M)\mathrm{T}(N\times M)R$ actual footprint on silicon. There is, consequently, a strong interest in improving passive crossbar dimensions above the 64×64 size report so far[53].

Box 1: Analysis of non-ideal parameters of RS memories that impact neural network accuracy

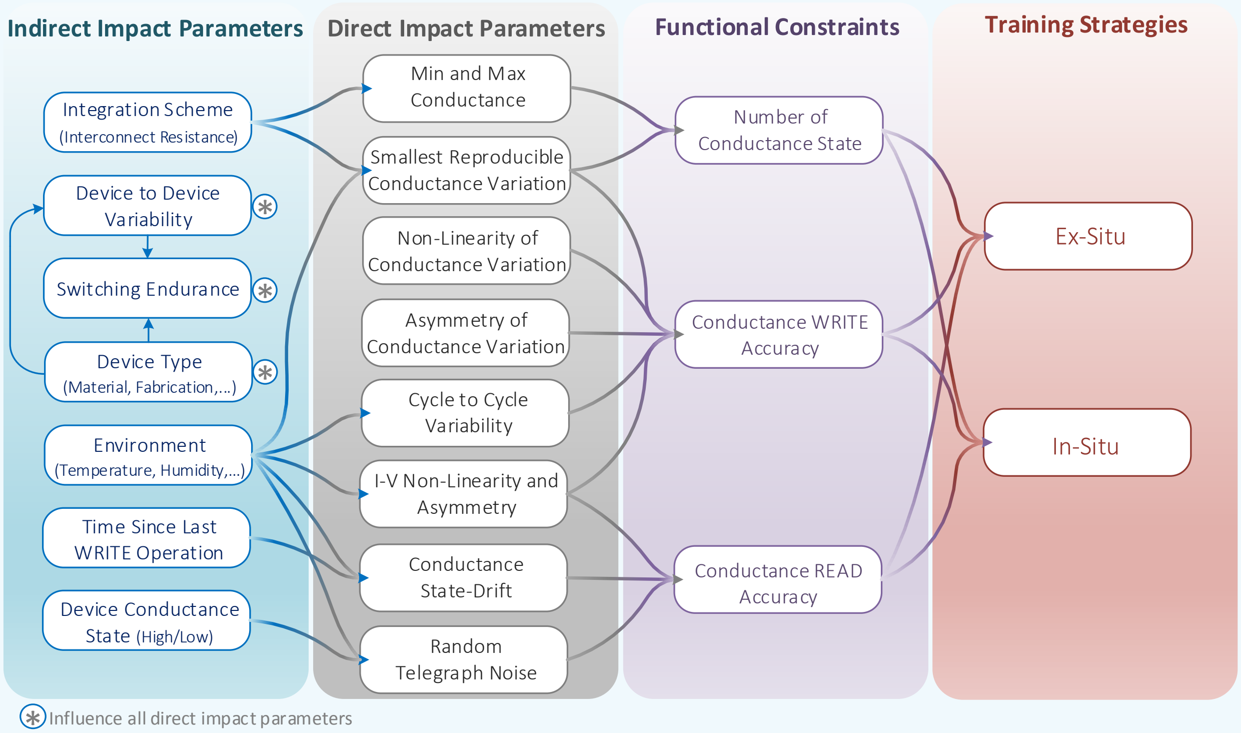
Indirect Impact Parameters
Direct Impact Parameters
Functional Constraints
Training Strategies
Integration Scheme (Interconnect Resistance)
Device to Device Variability
Switching Endurance
Device Type (Material, Fabrication,...)
Environment (Temperature, Humidity,...)
Time Since Last WRITE Operation
Device Conductance State (High/Low)
Min and Max Conductance
Smallest Reproducible Conductance Variation
Non-Linearity of Conductance Variation
Asymmetry of Conductance Variation
Cycle to Cycle Variability
I-V Non-Linearity and Asymmetry
Conductance State-Drift
Random Telegraph Noise
Number of Conductance State
Conductance WRITE Accuracy
Conductance READ Accuracy
Ex-Situ
In-Situ
⊛ Influence all direct impact parameters

Figure 4. Schematic classification of a memristor-based system's non-idealities according to the way they impact Artificial Neural Network (ANN) accuracy. Each arrow connection should be read as *"could have a significant influence on"* but with no consideration for their relative impact level. The first column *Indirect Impact* can be considered as hyper-parameters that only impact the ANN accuracy through their influence on other parameters. The second column *Direct Impact* represents the fundamental parameters that directly influence the ANN accuracy. The third column *Functional Constraints* lists some measurements that are often used as reference to quantify a memristive device performance. And the last column *Training Strategies* contains the two main approaches to train a network on RS memories.

Designing a RS-based system compatible with established microelectronic industrial technologies and large-scale production is only one part of the challenge. Since RS devices have inherent physical imperfections [64–68], it is necessary to find efficient ways to deal with them. The impact level of such non-ideal parameters can be varied on different applications and here we focus on how they influence VMM-based ML applications, specifically, the accuracy of physically implemented Artificial Neural Network (ANN). The accuracy of an ANN denotes the output success rate for a task for which it has been trained. For example, the accuracy of digit recognition using the MNIST database corresponds to the proportion of correctly classified image from a test dataset. In the context of RS-based ANN, we can distinguish two training strategies: in-situ and ex-situ[69]. In the in-situ scheme, the training is performed directly on the hardware by updating weights (i.e. the conductance of all devices) after each training epoch. This approach is notably impacted by all device non-ideal parameters that affect the conductance writing accuracy[70–73] (Figure 4) because this operation is repeated several times during in-situ training. In the case of ex-situ, the weight matrix is initially calculated in software ANN before to be transferred to the device array by encoding the determined weights into the conductance for each cell. In that scope, the conductance programming process occurs only one time per device, which make it viable to apply advanced methods to mitigate non-ideal parameters related to writing [52,70]. Finally, a hybrid strategy showed some interesting results by fine tuning the network weights after the transfer[54]. To better understand the different impacts of RS-based system non-ideal parameters on training strategies, it is interesting to not only consider their impact on functional constraints (write/read accuracy, latency, energy consumption, etc.) but also the inter-dependence between the different parameters.

For example, the switching endurance, which represents the average number of cycles before losing resistive switching behavior, directly impacts minimum and maximum conductance values over cycles[74], which in turn contribute to determine the total number of conductance state. Therefore, poor switching endurance could indirectly lead to low number of conductance state, or even failure such as stuck-at-fault where only one conductance state exists[75]. The impossibility to update the conductance decreases the ANN accuracy[56], even more so for ex-situ training where weights are supposed to be mapped on working devices. The same analysis can be made with the device to device variability parameter, which becomes a problem only if this variability concerns critical device characteristics like cycle to cycle variability[65] or the overall asymmetry of the conductance variation[70]. Further work should be conducted on the interactions between all non-ideal parameters in order to clarify their direct and indirect impact on the accuracy of physically implemented ANN, which could help the design and demonstration of mitigation strategies.

2.4 Scalability

In digital approaches, computational scalability of the Add operation is ensured by pipelining simple logical operations of single bits, thus allowing for very large vector-matrix manipulation (adding multiple dot-product, for instance). The digital approach is based on a trade-off between scalability of the operation, and computing time (e.g. how many clock cycles and basic operations are required). In RS-based Add operation, adding multiple dot-products is realized in a single time step. This advantage comes at the price of higher instantaneous power requirements. Adding currents from multiple dot-products results in a large current summation that could become a bottleneck for the VMM operation (Figure 5(a,c)). Adding infinite size of dot-products results in infinite time in the digital scheme and it results in infinite power for Kirchhoff's law-based approach. Practically, memristor-based VMM has been reported for a matrix size of up to 128×64[14]. While this was demonstrated with pseudo-crossbar having micron size electrodes, such limitations in matrix size should become a serious computational scalability challenge with electrodes in the tenth of nanometer range that would prevent sinking large currents through them. The 64×64 VMM operation was demonstrated in[53] using a purely passive crossbar with a more advanced patterning process (<200 nm). Dot-product demonstration with other integrated approaches[76,77] are today limited to small vectors dimensions, with a vector dimension below 25, and they impose restrictions on the VMM application. There is also a concern that this limitation will get worse by decreasing the metal line width and will require high aspect-ratio lines to achieve a high conductivity interconnect[60]. Alternatively, increasing the mean resistance of RS devices would increase scalability significantly by reducing power consumption at the cost of lower VMM operation speed. The inference operation speed is determined by the delay induced from the input circuits, RS-based crossbar array, and output circuits. In very large RS arrays, there are several parameters that should be considered to determine the delay such as: interconnect resistance; interconnect capacitance; RS cell resistance; overhead circuit's impedance and capacitance. The inference delay is calculated based on the Elmore delay model as follows,

$$t_{\text{inf}} = t_{\text{settling}} + \sum_{i=1}^{4} \tau_i, \qquad (1)$$

where t_{settling} is the settling time of the output circuit. As it can be seen in Figure 5(b), the parameters τ_1, τ_2, τ_3 and τ_4 are the delays from row, RS cell, column and output circuit, respectively[78]. By considering the LRS resistance of the device much larger than the interconnect resistance between each two adjacent cells, the delays τ_3 and τ_4 are dominant in very large arrays. By increasing the LRS of the RS cell, the inference time delay increases as it is impacting both τ_3 and τ_4. Therefore, the throughput of the system will be reduced accordingly. However, increasing the size of the array would also impact the inference delay e.g. increasing the number of rows will make τ_3 the dominant term to impact the total delay and it will slowly increase the delay. On the other hand, increasing the number of columns will increase the latency. Crossbar and pseudo-crossbar scalability challenges can also be related to the computing performance (e.g. accuracy). Unlike digital approaches where input digital signals margins allow to cope with noise and parasitic, analog VMM implementation accuracy is negatively affected in the case of large vector operations. The resulting mismatch between the resistance of the memory cells and the one of metal interconnects becomes critical in large crossbar arrays (Figure 5(d)). The same bias applied to the word-line is seen differently by each cell in the crossbar due to linear voltage drops that lead to a decrease of accuracy for the VMM operation. A straightforward physical solution to these constraints is to limit the size of the crossbar array and thus the VMM performed in one step. Note that small VMM dimensions are largely used for convolutions in Convolutional Neural Network (CNN). In conclusion, scalability of memristor-based VMM operation represents a future research direction that requires innovative solutions at both technological and system levels.

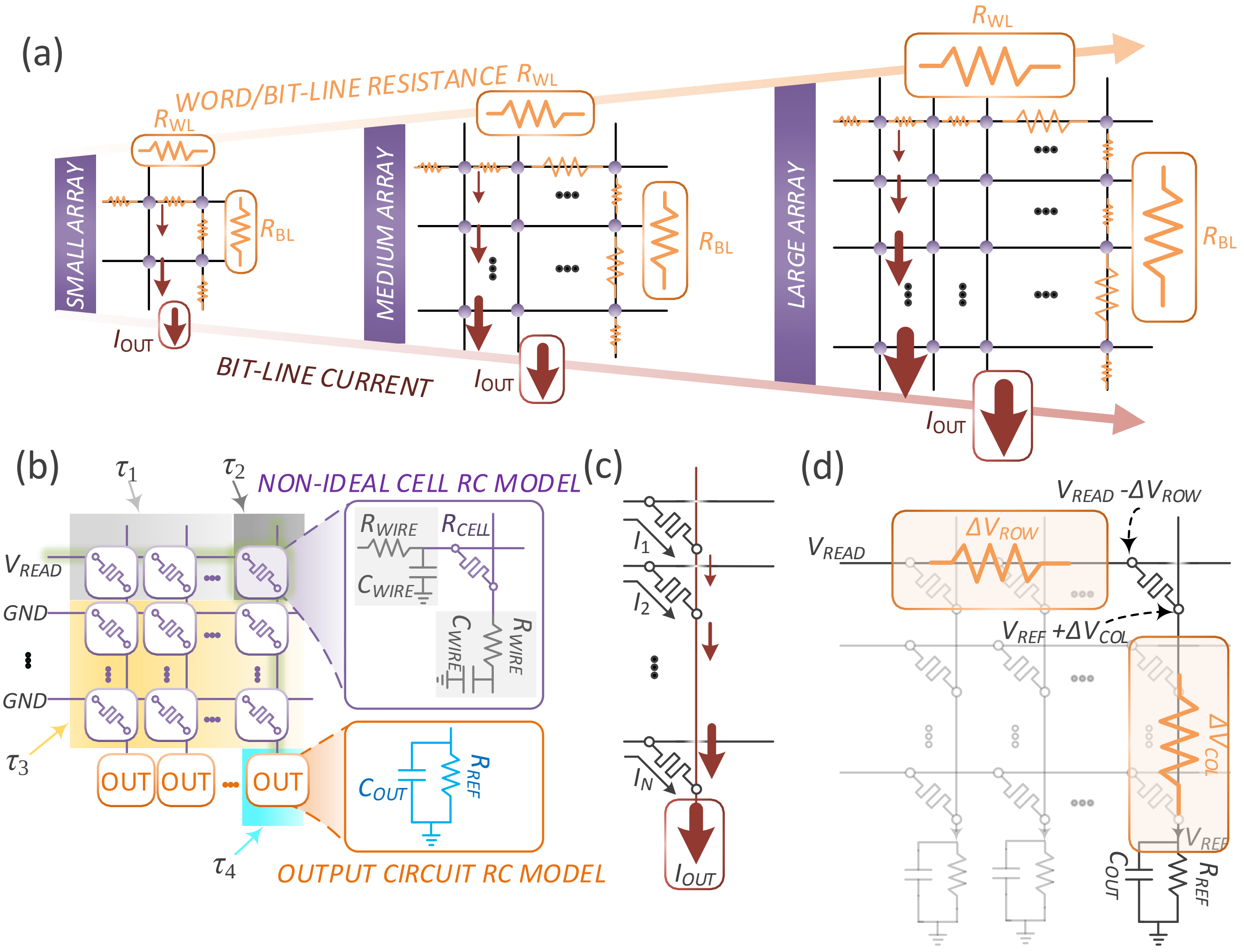

Figure 5. Scalability challenges and RC network Elmore delay model for RS crossbar array. (a) The scalability challenges: Both bit-line current and word/bit-line resistance increase with the size of the RS crossbar array. (b) RS crossbar RC Elmore delay model is shown by dividing the array delay into four regions corresponding to τ_1, τ_2, τ_3, and τ_4 which are the delays from a row, RS cell, column and output circuit, respectively. (c) Increasing the number of rows increases the accumulated current in the column and can become a major challenge for output circuits design. The same limitation applies for the large number of columns required to inject large current into the row and affecting input circuits design. (d) The line resistance is another challenge for scalability of RS-based arrays due to the voltage degradation in the rows (ΔV_{ROW}) and columns (ΔV_{COL}). This issue can be leveraged by engineering optimization and/or compensated by input/output circuits strategies.

3 Circuit design challenges for VMM implementation

3.1 Background

As mentioned previously, projected energy consumption for a single dot-product operation can indeed be as small as 0.01 fJ, while 0.2 pJ are consumed with 8-bit digital VMM based on 45nm CMOS technology node[2]. However, this comparison is not a complete picture since it does not consider energy consumption for input/output signals generation. A more rigorous evaluation of memristor-based dot-product energy consumption should be done by considering 8-bit digital-to-analog converter (DAC) at the input and 8-bit analog-to-digital converter (ADC) at the output where both components consume approximately 0.1 mW and can be run at the frequency of 1 GHz (1 ns clock cycle). The total energy required to compute the 8-bit dot-product with RS devices becomes largely dominated by these DAC/ADC-based overhead circuits since $E_{DAC} + E_{ADC} = 2 \times 0.1 \times 10^{-3} \times 10^{-9} = 0.2$ pJ. This simple example therefore highlights the importance of the overhead circuitry in the assessment of VMM engine

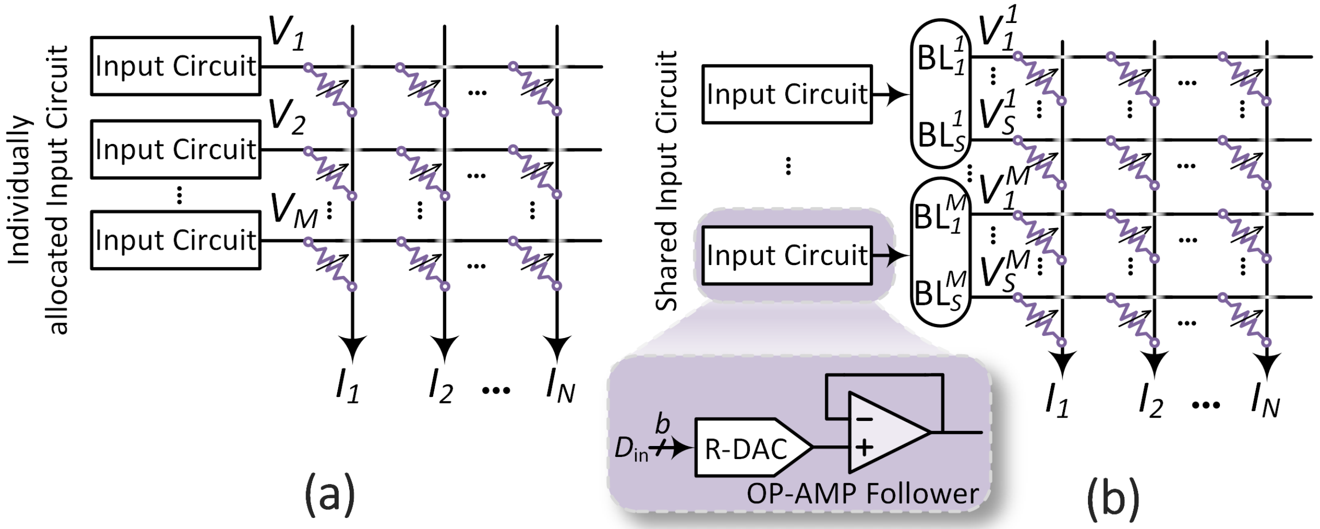

Figure 6. (a) General RS-based current-mode VMM architecture with individually allocated input circuit for each word-line. (b) Shared input circuit architecture is displayed with a sample schematic of resistive DAC with OP-AMP follower.

performance. While most of the approaches so far have been using software-emulated or custom on printed circuit boards (PCB), there are recently only a few fully integrated chip demonstrations. These demonstrations benefits are two-fold: (1) Exploring CMOS design overhead circuits and their compatibility with RS devices; and, (2) Exploring various strategies at the system level for building a fully operational chip. These choices are defining the application field of the VMM engine and impacting both the energy and accuracy performances.

3.2 Input circuits

VMM engines are mostly envisioned to boost energy and speed performances of conventional hardware (CPU and GPU) for specific tasks such as image compression, machine learning algorithms, combinatorial optimizations or solving linear and partial differential equations. In these applications, the VMM operation needs to be integrated into a digital environment used to manage the higher order functions such as data management and VMM definition/programing. Generating an analog input voltage from digital input data can be implemented with Digital-Analog Converters (DAC), which implies a trade-off between DAC's resolution and energy consumption. Generally, current-mode VMM architectures are utilized for higher resolution VMMs and each word-line is connected to a single input circuit such as external voltage-mode DAC (Figure 6(a)). However, sharing DAC circuits by providing binarized input voltage to multiple word-line (Figure 6(b)) is another design option to avoid using a power hungry and spacious high-resolution DAC for each word-line. Since dot-product operation is limited to 8-bit by the RS conductance available states, there is no interest in using DACs with resolution higher than 8-bits. However, using high resolution DAC circuits will result in higher cost and reducing area and power efficiency of the VMM platform. For RS-based VMM engines, the foremost parameters used for describing the performance of the DAC are area, power consumption and, more importantly, the output impedance as it limits the number of memristors that one DAC can drive. In other words, the maximum output current is bounded by the DAC output impedance for a given voltage supply. The following describes an analysis method regarding the trade-off among essential DAC parameters for VMM engine applications. This method analyzes the design trade-off of a high-resolution digital-to-analog converter (DAC) with low output impedance, which is a resistive DAC with an operational amplifier (OP-AMP) follower output stage Figure (6(b)). A similar approach can be used for estimating the design trade-off among bandwidth, resolution, die-area, and power consumption for a DAC with a different architecture. The most power-hungry blocks in the DAC are: (1) the analog circuitry that is used for driving the memristor devices; and, (2) The digital circuitry that is used for storing the data and distributing the clocks. The power dissipation of the DAC can be divided into the switching/leakage power of the digital circuit, and the static/dynamic power of the analog circuits. The power dissipation of digital circuits can be estimated by,

$$P_D = f_{2b} C_p V^2 + P_{Leakage} \tag{2}$$

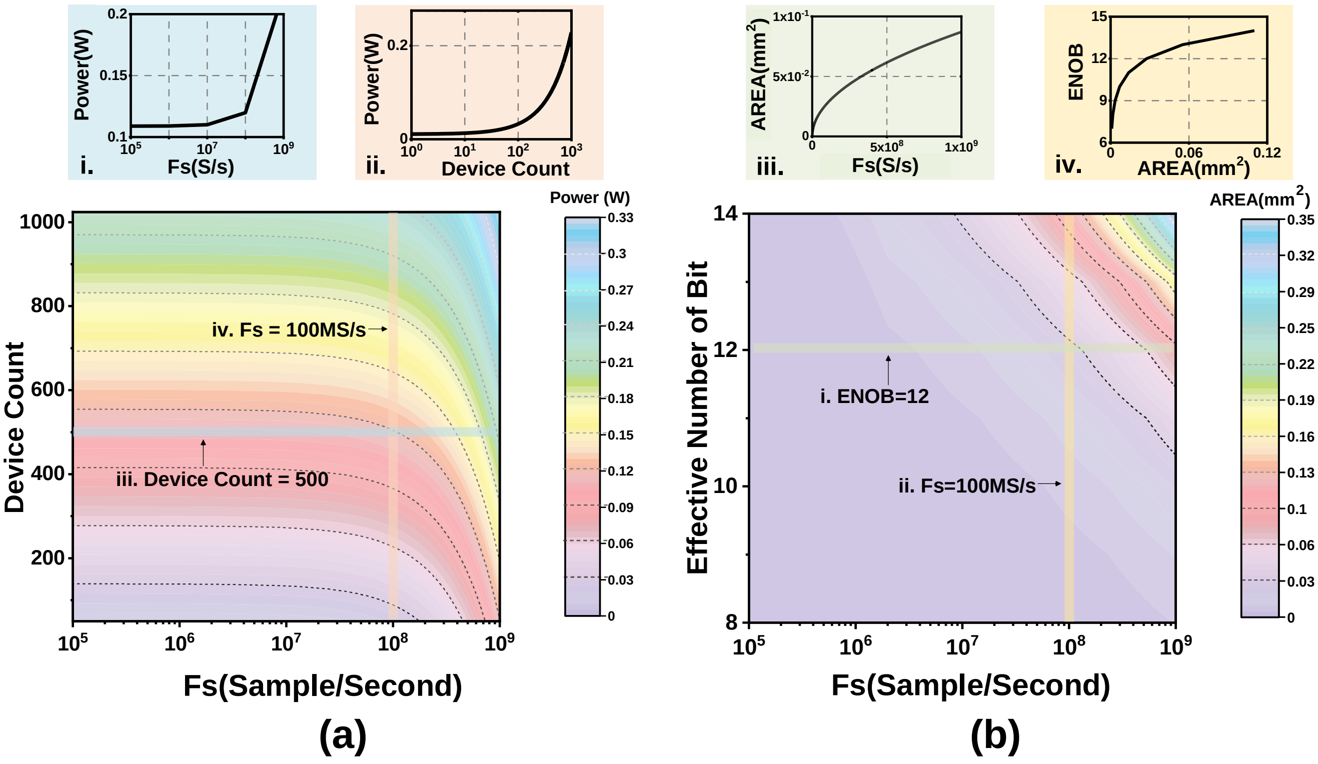

Figure 7. (a) The DAC area usages versus the sample frequency and effective number of bits. In the following two specific cases has been described by sub figures. (i) Area usage versus sample frequency at effective number of bit (ENOB) equals 12. (ii) Area usage versus ENOB at sample frequency of 100 MS/s. (b) The DAC power consumption versus sample frequency and device count (the number of memristor devices driven by one DAC), assuming the total parasitic capacitance is 10 pF. In the following two specific cases has been described by sub figures. (iii) The power dissipation versus sample frequency when the device count is 500. (iv) The device count versus power consumption at sample frequency of 100 MS/s.

where f_{2b} is the DAC maximum output frequency that equals twice the bandwidth, C_p is the total parasitic capacitance, V is the supply voltage and $P_{Leakage}$ is the leakage power that depends on technology node(around several pico-Watts for an inverter in 65 nm technology from 1V power supply). For resistive DAC the main analog power is from the OP-AMP follower output stage, which usually employs a class-A output stage that has a maximum power efficiency of 50%. So, the analog power can be estimated using,

$$P_A = n \times V^2/R, \tag{3}$$

where n and R parameters are the number of devices which have been drived by the DAC and minimum RS device resistance, respectively. Assuming the minimum resistance of each RS device is 50 kΩ, and the power supply voltage is 3.3 V,the estimated power consumption is shown in Figure 7(a). It has been shown that the power consumption is almost proportionate with number of devices below 100 MS/s (Mega sample per second) operating frequency and this is because the analog power is dominating when the quantity of RS devices becomes relatively large. While in higher operating frequencies than 100 MS/s the power consumption will be impacted mainly by operating frequency rather than number of devices when digital power becomes a dominant term.

The die-area is mainly constrained by the needed DAC resolution that limited by the element matching and noise. For resistive DAC, the major noise is from the amplifier at the output stage, and input-referred noise is given,

$$V_{rms}^2 =\cong \frac{4K}{C_{ox}WL}\ln\left(\frac{f_2}{f_1}\right) \tag{4}$$

where W and L are the width and length of the input pairs, respectively. Parameter K is Boltzmann's constant, C_{ox} is the gate capacitance per unit area, and f_1 and f_2 are the low corner and high corner frequencies, respectively[79].

The matching of the resistor is described as follows,

$$S_R = \frac{1}{W_R\sqrt{R}}\left(k_a + \frac{k_p}{W_R}\right) \tag{5}$$

where W_R is the width of each resistor and R is the resistance, and k_a and k_p are the constants that highly depend on the technology representing the contributions of areal and peripheral fluctuations[80]. Figure 7(b) shows the estimated area of resistive DAC versus the operating frequency and the effective number of bits. The area is changing almost linearly with the operating frequency and exponentially with the effective number of bits. A similar approach can be used for estimating the area and power consumption for the DAC with a different architecture. In addition to undesirable high energy consumption of the high-resolution DACs, delivering perfect analog input signal on each memory cell is challenging since it can be easily deteriorated by crossbar arrays imperfections. As mentioned previously, voltage drop along the metal lines (Figure 5(c)) induces analog values distortion (each resistive memory from a line will be subjected to analog voltage drops when the distance from the input circuit increases). This issue can be solved by additional computing overhead via software processing of the data as proposed by[72]. In this approach, voltage drop along the metal lines is calculated and compensated by RS conductance adjustment. Another limitation affecting the VMM accuracy when the input data is encoded with the analog voltage amplitude signals is the non-linearity of the current-voltage characteristic of RS elements. In this case, the actual conductance of the RS element is input-dependent and can impact the VMM resolution. This problem could again be tackled by data pre-processing including the effect of RS devices' non-ideal parameters into the analog input but can become quickly very complicated if high variability in RS device is to be integrated in pre-processing. Alternatively,[76] proposed a method to solve this limitation by encoding the analog input signal with pulse width modulation. This strategy comes at the cost of multiple clock cycles for each encoded input but mitigate I-V non-linearity. In this chip, each channel includes one read DAC, and two write DACs as input circuits. A digital controller converts a 6-bit input into an n-element pulse train of identical Return to Zero (RTZ) pulses where n is the input data. The digital output from controller drives a 1-bit DAC, which delivers a pulse train of read-voltage pulses to the crossbar row. An advantage of using RTZ pulses is that the non-idealities introduced at pulse transitions are proportional to the input and show up as a gain error that can be canceled in software. Finally, digital-analog conversion can be also avoided by using analog inputs in their digitized form. Each bit from the analog input number is computed sequentially from the least significant bit to the most significant one. This strategy will increase the number of operations to compute a single VMM but will preserve the analog resolution.

3.3 Output Circuits

Output signals from a RS-based VMM operation are analog currents that needs to be converted into digital numbers. A straightforward solution is to use ADC and Trans-Impedance Amplifiers (TIA). The ADC resolution depends directly on both the conductance resolution of each RS element and the VMM size. For example, 1-bit RS conductance with a vector dimension of 256 (256 lines connecting to one bit-line) requires at least 8-bit of resolution to discriminate all output levels. 5-bit RS memories with the same vector dimension requires a 13-bit ADC, which represents, in itself, a serious design challenge to preserve energy consumption/area efficiency. Employing high resolution ADC in such arrays is one option for distinguishing the analog output levels, which requires a careful cost and overhead analysis. Many parameters are used to assess the performance of an ADC such as: input impedance; supply rejection; metastability rate; power consumption; die area; signal to noise and distortion ratio (SNDR); and, etc. [81]. In a typical RS-based VMM engine, the most important metrics to consider for the ADCs are their resolution, their sampling frequency (f_s), and their surface area on the die that affects accuracy, throughput, and cost, respectively. Figure 8 reveals the main aspects trade-off of the ADC published in the International Solid-State Circuits Conference (ISSCC) from 1997 to 2020. The technology node is the fundamental factor that constrains the area of an ADC (Figure 8(f)), whereas a survey of state-of-the-art ADCs [82] reveals that, for a smaller technology node and more diminutive voltage supply headroom, the power consumption is usually bounded by the thermal noise so that one added bit demands quadrupled power rather than only proportional fCV^2. The ADCs with higher resolutions are slower and less power-efficient (Figure 8(c)) while the ADCs with higher sampling frequency have worse energy-efficiency and lower resolution (Figure 8 (b,d)). The achievable performance of the ADC can be predicted by two well-known figure of merits (FOM) [83–85] as:

$$FOM_S(\mathrm{dB}) = SNDR + 10\log_{10}\left(\frac{ERBW}{P}\right), \tag{6}$$

where *ERBW* is the bandwidth of the ADC, *P* is the total power dissipation.

$$FOM_W(\text{fJ/CONV.STEP}) = \frac{P}{2^{ENOB} \times \min(f_s, 2ERBW)}, \tag{7}$$

where *ENOB* is the effective number of bits. In general, the achievable best FOM_S is decreasing along with the increasing of frequency, e.g. doubling f_s or increasing 1-bit resolution postulates quadrupled power consumption (Figure 8(e)). In addition, reducing FOM_W demands an increase of die-area, e.g. 50% power reduction or 1-bit more resolution need 25% more die-area (Figure 8(f)). Overall, the choice of ADC architecture depends on the needs of the application. If each memristor crossbar word-line or bit-line requires one high-resolution ADC (>10-bit), successive approximation register (ADC) or delta-sigma (DSM) ADC can be utilized as (SAR) and DSM have slightly smaller form factors (Figure 8(a)) and significantly better SNDR. Voltage control oscillator (VCO) based ADC or SAR ADC are more suitable to smaller technology node implementation since they do not rely on high gain/bandwidth amplifiers that are limited by intrinsic transistor gain [86]. If the inference operation takes longer than 10ns, low resolution/high-speed flash ADC can be applied via time-multiplexing to minimize die-area since an 8-bit ADC is usually needed for a typical neural network to achieve more than 90% classification accuracy [76,87]. The best possible ADC performance can be estimated based on the system requirement. A decent system-level design can reduce the needed performance of the ADC significantly. The dashed line shown in Figure 8(b) marked the lowest possible ADC power consumption for a given sampling frequency, and the dashed line shown in Figure 8(c) marks the maximum possible ADC SNDR for a given power consumption limit. Therefore, the trade-off among speed, power, and accuracy of ADC can be described by the following equation,

$$P = FOM_{W,\min} \times 2^{ENOB} \times f_s, \tag{8}$$

where $FOM_{W,\min} = 2 \times 10^{-15}$ for the best state-of-art ADC designed in 28 nm technology. The relationship between the peak SNDR and ENOB is,

$$SNDR_{\max}(\text{dB}) = (ENOB \times 6.02) + 1.76. \tag{9}$$

The dashed line in Figure 8(d) labels inevitable trade-off between the peak SNDR and sampling frequency within the current state-of-art ADC

$$SNDR_{\max}(\text{dB}) = 165 - 10\log_{10}^{ERBW}. \tag{10}$$

Figure 8(f) reveals the trade-off between the energy efficiency matrix and area efficiency. The area and energy efficiency improves with shrinking the technology nodes, while they are roughly bounded by the following relationships,

$$Area(\text{mm}^2) = A - (10^{-4} \times FOM_W). \tag{11}$$

where A is the technology depended factor that equals to 2×10^{-3} for 14 nm technology. The analysis shown above is used for estimating the performance of relatively low-resolution ADC ($<$ 14-bit). For higher resolution ADC ($>$ 14-bit), adding one more bit means increasing 6 dB SNDR, quadrupled less noise power and four-fold larger overall capacitance, as the thermal noise at the input of the ADC equals to KT/C (where K is the Boltzmann constant, T is the temperature, and C is the capacitance at the input of the ADC). This relation is well defined by Shreier's FOM[83]. Figure 8(e) shows the relationship between the Shreier's FOM and sampling frequency, the maximum achievable FOM at low frequency ($<$ 10 MHz) is 192 dB, and at higher frequencies ($>$ 10 MHz), the best achievable FOM equals to,

$$FOM_{S,\max}(\text{dB}) = 192 - (10 \times \log_{10}^{ERBW/10}). \tag{12}$$

An alternative sensing approach is to replace the TIA block by a charge-based accumulation circuit. This strategy was used to cope with pulse width modulation encoding that excludes the utilization of TIA[76]. Note that the same approach could be used along with other encoding techniques such as digitization of inputs and pulse amplitude modulation. To maintain precision of the RS-based VMM hardware, the same trade-off in the ADC resolution with crossbar array size is applicable and requires a design optimization in terms of energy consumption and footprints.

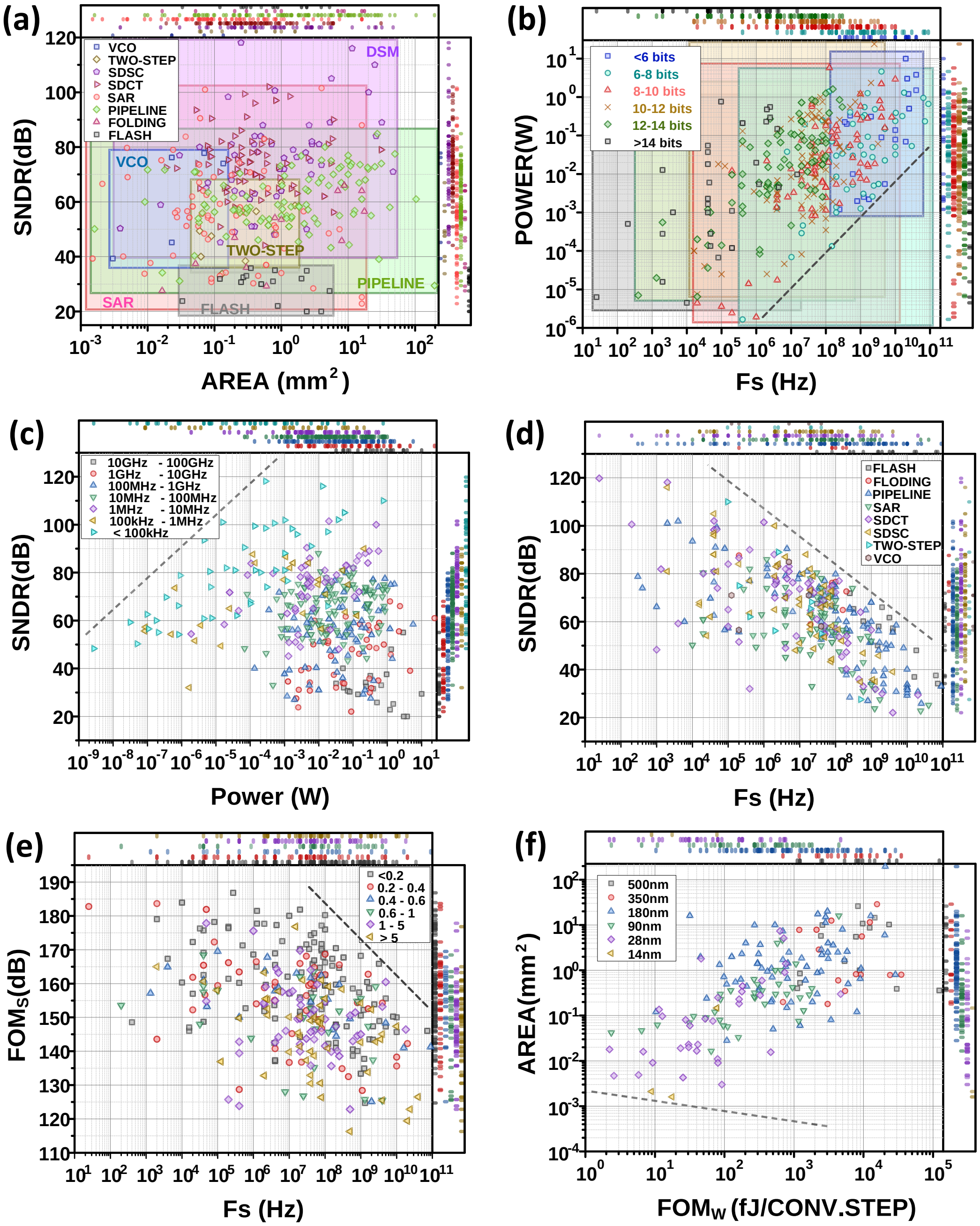

Figure 8. Main aspects trade-off of the ADC published in the International Solid-State Circuits Conference (ISSCC) from 1997 to 2020.(a) ADC area increases with SNDR and is classified based on different ADC architectures. (b) ADC power consumption increments with the increase of Nyquist sampling frequency and marked based on different resolutions. (c) ADC power consumption versus the SNDR has been illustrated for different design operating frequencies. (d) ADC sampling frequency versus the SNDR has been shown for different ADC architectures. (e) ADC FOM versus sampling frequency is depicted and classified based on CMOS technology node (in μm). (f) ADC area versus FOM is shown and categorized based on different CMOS technology node.

3.4 Recent Chips Demonstration on Integrating CMOS Circuits and RS devices

Implementation of VMM hardware using in-memory computing property of RS-based array has become a topic of interest for AI hardware research groups in recent years. Some of these efforts [14,88] have used discrete integrated circuit components connected to the RS array and they did not present a complete integrated system in a single chip.

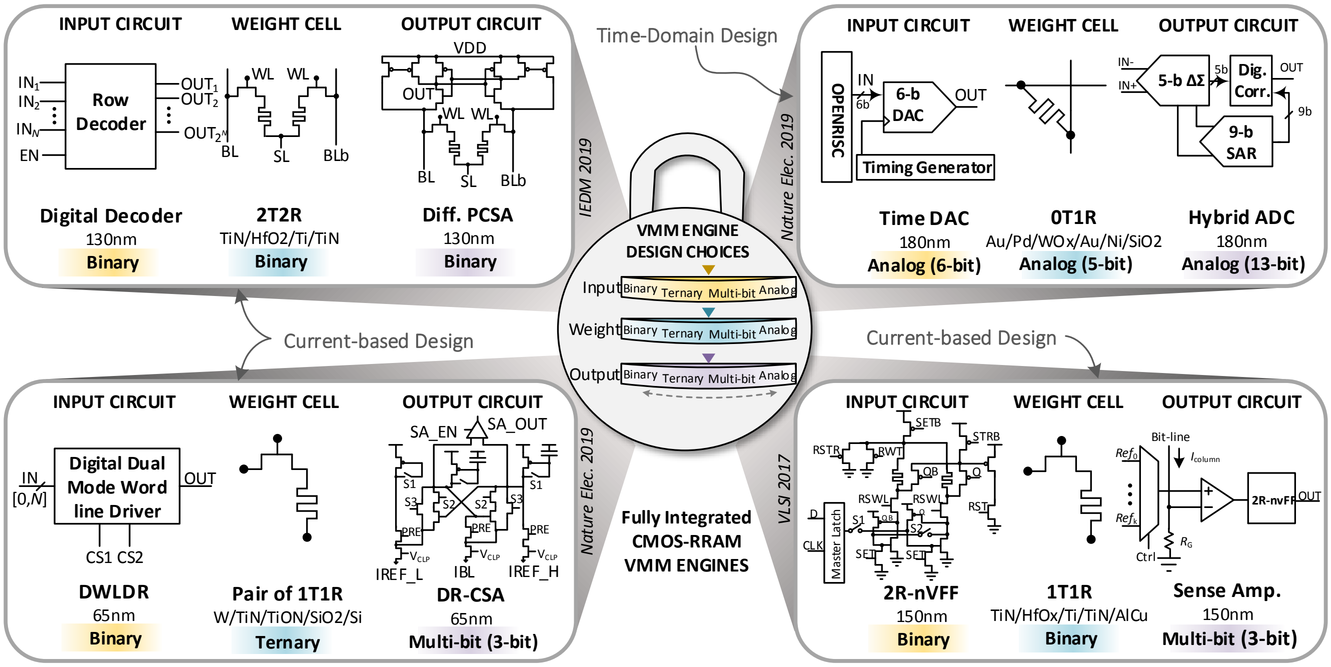

Figure 9. Design choices for RS-based VMM engines are defined based on the combination of the input, weight cell, and output precision targeted for specific applications. Here, a combination lock is shown as VMM design choices, which may be unlocked with different combination of the input, weight, and output. The input, weight and output choices are binary, ternary, multi-bit, and analog input. Here, we have depicted four examples of different design choices from the recent fully integrated CMOS/RS-based chips [76,77,89,90].

However, there are few fully integrated CMOS/RS devices chips implemented for VMM-based applications. These fully integrated VMM engines can be categorized into various design choices based on the precision of the selected weight, input and output. However, this categorization can be complemented by considering the classification of these platforms into current-based and time-domain designs. As can be seen in the Figure 9, choices for input, output and weight cell includes binary, ternary, multi-bit, and analog. However, selecting a design choice is directly depending on the target application requirements and its functional aspects e.g. accuracy level, speed and etc. There have been several device-, circuit- and system-level concepts proposed to enhance the efficiency and functionality for each of these design choices.

As an example, for a binary weight cell design with a circuit level proposition for input and output circuits, a non-volatile intelligent processor (NIP) [89] has been designed by using 4 kb 1T1R binary HfOx-based cells and using 150 nm CMOS technology. This work proposes a non-volatile flip flop circuit by integrating two RS cells into its design for the input and output sensing blocks to avoid high cost DAC and ADC blocks. The output sensing circuit has an adaptive design and can support from 1-bit to 3-bits of resolution. This design improves energy and area efficiency by eliminating the data conversion circuits overhead and turning off the unwanted cells by input-controlled access transistor scheme in 1T1R array. The other physically implemented chip is a binary VMM engine presented in[90] by using 2T2R differential weights with input-controlled access transistor scheme and a pre-charged sense amplifier (PCSA) circuit. This chip was developed for binarized neural network demonstration but consists, essentially, in a binary dot-product operation. The 2 kb HfOx-based RS devices have been integrated on top of the forth metal layer in CMOS 130 nm technology node. The PCSA circuit is differential and connected to the both bit-lines of the 2T2R cells in each column. Due to the binarized neural network properties[91], the weights and activation functions are binary and there is no need for multipliers. This design is very efficient for in-memory computing applications where activation functions are implemented by XNOR gates and additions are carried out by popcount gates. This chip is purely digital and it is free from any D/A or A/D conversion that results a high energy and area efficiency performance.

In addition to the mentioned design choices, for ternary weight design, a 1 Mb 1T1R array and its CMOS peripheral circuits were integrated on a single chip in 65 nm CMOS technology node[77]. This implementation proposed new circuit peripherals and architecture level idea to enhance the area and energy efficiency. This platform implements configurable logic operations (XOR, AND and OR) in addition to inference operation. Binary inputs and ternary weights are implementing inference with positive and negative weights located in two separate sub-arrays. Partial MAC results computed from each sub-array are added together to compute a partial MAC. To avoid using costly DAC circuits, this work proposes Dual Word Line Driver (D-WLDR) circuit to apply inputs in both memory and inference modes. These circuits include small digital buffers occupying small area and fitting with the pitch size of the 1T1R cell in the word line. To overcome the issue of area efficiency due to high precision ADC blocks and to enable a highly parallel inference operation, small offset current mode sense amplifier (ML-CSA) and input-aware reference current generator circuit (MIA-RCG) are proposed. MIA-RCG is generating various reference currents in reference arrays to increase the bit-line signal margin between different states for each mode of operation (logic or inference). ML-CSA is minimizing the offset in sense amplifier due to the mismatch of CMOS devices in the bit-line. To further, enhance the readout accuracy and tolerance for a small read out margin, Distance Racing Current Mode sense amplifier (DR-CSA) is proposed as it shows an improvement in sensing margin by two times in comparison with the mid-point sensing scheme. The platform demonstrates a promising energy efficiency and inference accuracy for various precision values (1-, 2-, and 3-bit), but with limited array size (VMM is limited to dimension 12). In [92], a 158 kb VMM engine is designed in 130 nm CMOS technology and it is tried to mitigate the issue of: large sensing current in the columns, ADC circuit overhead, and the problem of voltage drops and transient error of MAC operation in large VMM. A signed weight 2T2R cell has been used in order to reduce the column's sensing current by getting benefit from the differential current. In this work, a quasi-3-bit weight (7-level) is used by positive and negative 1T1R cells that locally cancels their current in the shared column and this should fairly solve both problems of large sensing current and voltage drop impact. This work also presented a low power adjustable resolution ADC circuit (LPAR-ADC) which is reconfigurable from 1-bit to 8-bit precision. The integration and quantization scheme in LPAR-ADC suppressed overshoot and fluctuation of the sensing current improves the transient error due to the sensing stage. The proposed VMM engine is providing a high energy efficiency of 78.4 TOPS/W when sensing the output by 1-bit precision and high inference accuracy around 94% for MLP of MNIST classification task with 8-bit sensing precision in both ADC stages of the network. For multi-level weight design choice, [54] proposed a hardware implementation of CNN using 1T1R RS-based VMM engine in 130 nm CMOS technology node. In this hardware, eight 2 kb processing element (PE) chips have been integrated on a custom designed PCB to implement a five-layer CNN network. Each of these PE chips, in addition to the RS-based array, includes switching matrix circuits for input and output, 8-bit ADC, and shift and add blocks. 4-bit differential pair of 1T1R cells is deployed as weights by tuning the 8-level RS devices. Analog inputs are encoded into 8-bit binary sequential pulses in eight time-intervals and applied via external voltage generator to PE chips. Each PE chips include four ADC blocks with 8-bit precision to sense 128×16 RS array. Each ADC block is shared between four columns by sample and hold (S/H) circuits for time multiplexing to reduce the overhead cost of the analog to digital conversion. To reduce the latency of inference, each of these four columns are connected via a pair of S/H blocks. In first inference step, one S/H block in each pair is sampling the output of its corresponding column. During the next inference step the other S/H block in each pair samples the output while the ADC carries out sensing of the output from all four blocks that sampled in the previous inference cycle (first inference output). This inference scheme reduces the inference latency by pipelining the computation. The hybrid training scheme is utilized to avoid accuracy loss due to the device- and array-level imperfections. This was done by mapping the ex-situ weights on all PE chips in initial steps and, subsequently, applying multiple runs of in-situ learning on the shared fully connected layer PE chips. This VMM engine design has a very high computational efficiency (1.164 TOPS/mm^2) and energy efficiency (11 TOPS/W) and it enhanced the inference accuracy for MNIST classification task up to 95.57%. First demonstration of VMM engine with analog weight deploying a passive RS crossbar by the size of 54×108 monolithically integrated with CMOS in 180 nm technology node on a single chip is presented in [76]. In this work, a charge-based inference is targeted to overcome the I-V non-linearity of the RS devices. In this context, the analog input is encoded by applying the discrete-time pulse train with the fixed-amplitude into a 6-bit time-domain DAC. The DAC then applies the corresponding 6-bit width modulated input pulse into the array. The bit-line accumulated charges are sensed by an incremental charge-integrating ADC. High resolution hybrid 13-bit ADC circuit is placed in both rows and column to enable bi-directional inference operation and it is comprised of a 5-bit first order incremental ADC, an 8-bit SAR ADC, and an additional 1-bit redundancy stage. The OpenRISC processor with 64kB SRAM along with timing generation blocks have been integrated in the chip to initiate different operation modes and control of the DAC and ADC blocks. High resolution input and output circuits and bi-directional inference capability make this platform highly flexible to implement different blend of machine learning applications. However, this flexibility adds cost as the number of ADCs is doubled beside the fact that high-resolution ADC consumes more power and area as

	Nature El. 2017[14]	VLSI 2017[89]	VLSI 2018[93]	Nature 2018[57]	Nature El. 2019[77]	ISSCC 2019[94]	ISSCC 2019[48]	IEDM 2019[90]	Nature El. 2019[76]	ISSCC 2020[92]	Nature 2020[54]
RS Device Type	RRAM	RRAM	RRAM	PCRAM	RRAM	RRAM	RRAM	RRAM	RRAM	RRAM	RRAM
CMOS Technology	2μm	150nm	40nm	90nm	65nm	55nm	130nm	130nm	180nm	130nm	130nm
RS filament Material	HfOx	HfOx	TaOx	-	SiOx	SiOx	HfOx	HfOx	WOx	TaOx HfOx	TaOx HfOx
Fully Integrated	NO	YES	YES	NO	YES	YES	YES	YES	YES	YES	YES
Crossbar Type	1T1R	1T1R	1T1R	3T1C + 2PCM	1T1R	1T1R	1T1R	2T2R	0T1R	2T2R	1T1R
# of Cells	8k	4k	4M	524k	1M	1M	18k	1k	6k	158.8k	16k
Weight Precision	Analog 6-bit	Binary 1-bit	Analog 4-bit	Analog >8-bit	Ternary	Multi-bit 3-bit	Multi-bit 2.3-bit	Binary 1-bit	Analog 6-bit	Multi-bit 3-bit	Analog 4-bit
Input Precision	Analog	Binary 1-bit	Binary 1-bit	Analog 9-bit*	Binary 1-bit	Binary 1-bit	Binary 1-bit	Binary 1-bit	Analog 6-bit*	Binary 1-bit	Digitized 8-bit
Output Precision	Analog 8-bit	Multi-bit 3-bit	Binary 1-bit	Analog 8-bit	Multi-bit 3-bit	Multi-bit 3-bit	Digitized 16-bit	Binary 1-bit	Analog 13-bit	Analog 1-8-bit	Analog 8-bit
Area (mm^2)	10.9	3.69	2.71	5.8	6	7.5	11.25	0.2	61.4	21.82	0.0708
SE (Mb/mm^2)	N/A	0.001	1.47	0.088	0.16	0.133	1.6	0.005	0.00009	0.0072	0.23
Throughput (TOPs)	1.64	0.101	0.66	20	0.019	0.012	0.78	0.0027	0.057	1.5	0.081
TOPs/mm^2	0.15	0.002	0.24	3.44	0.003	0.0016	0.069	0.013	0.0009	0.071	1.16
EE (TOPs/W)	119.7	0.462	66.5	27.4	16.95	53.17	1.65	4.2	0.1876	78.4	11
(TOPs/Wmm^2)	10.98	0.125	24.5	3.6	2.82	7.08	0.147	20.4	0.003	3.59	156

Table 1. Comparison of in-memory computing hardware with non-volatile memory blocks by considering capacity of larger than 1 kb.

well. Each of these VMM engine design choices offers different performance behavior and makes a trade-off between accuracy, energy efficiency, and area efficiency by considering the application constraints and demands that will be vital for the appropriate selection. The detail specification and performance of hardware implemented RS-based VMM engines is presented in Table 3.4.

4 System level development of RS-based VMM engines

4.1 Leveraging the cost of mixed analog/digital approaches and data trafficking

Performances of VMM engines appears to be strongly affected by the analog-to-digital and digital-to-analog conversion operations, even if the analog MAC operation by itself is very energy efficient. Note that this trade-off between in-memory computing of the MAC operation and overhead circuits cost should evolve favorably by increasing the dimensions of RS-based VMM engines. Indeed, as $N+M$ DACs and ADCs are required to drive a $N \times M$ crossbar array, the energy consumption and the analog/digital interface circuitry per operation should be thus decreased in the case of large-scale VMM engines. This is to be analyzed in the light of the important challenges that crossbar arrays scaling is facing (see discussion in section 3.3) and represents a vital point for the development of future RS-based VMM engines.

In the previous sections, we pointed out the important trade-off between in-memory computing of the MAC operation with the overhead circuitry required to drive the crossbar array. The proposed analysis considers only the potential improvement in terms of energy and speed offered by computing the MAC operation physically. It does not consider the energy consumption associated with data trafficking at higher levels, which has been identified in conventional computing platforms (e.g. GPU) as the most expensive operation. Moving data corresponds to both moving parameters of the MAC operation (e.g. matrix components) but also the input and output data (e.g. vectors to be computed / output vectors of the VMM). As can be seen in Figure 10(a-b), in the conventional von Neumann computing systems and near-memory computing (NMC) architectures all input, weight, and output data are moving between the processing unit and memory. However, the traveling distance in NMC systems are significantly smaller than the conventional von Neumann computing architectures. On the other hand, in-memory computing of the MAC operation is proposing to permanently store the matrix component into a dedicated non-volatile memory and thus reducing drastically the data movement for these parameters (Figure 10(c)). Nevertheless, I/O data still needs to be moved and can represent the main bottleneck of the overall system. Note that I/O data can also be used numerous times in the system for specific applications such as convolutional neural network and could benefit from

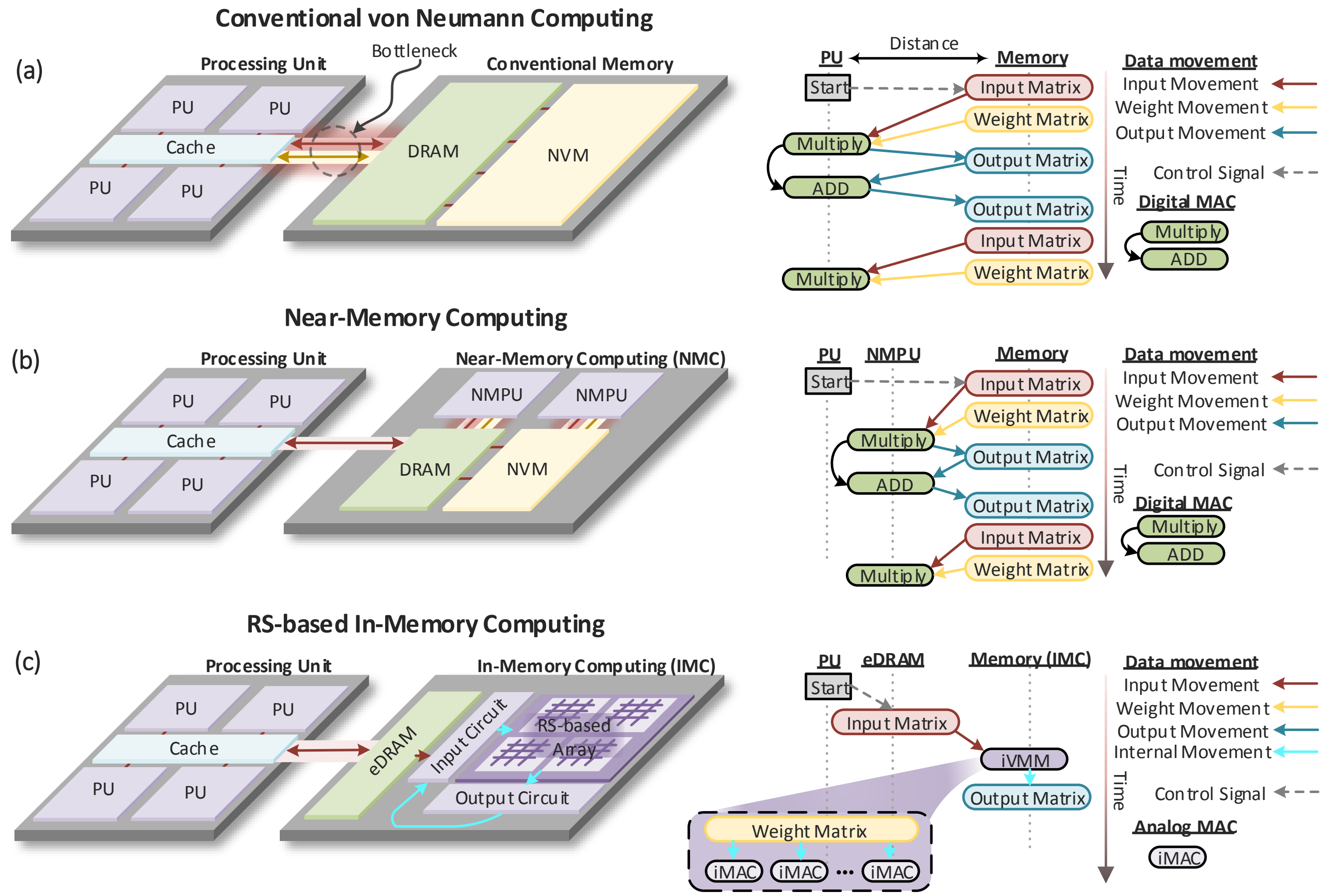

Figure 10. Three different computing architectures are shown with their corresponding data movement (input, weight and output) to carry out vector matrix multiplication operation. (a) Conventional von Neumann computing architecture is depicted comprising of processing unit and conventional memory. It has been shown that a high data movement for both inputs and weights as data needed to be fetched from or stored in the memory at different stages of the operation. Also, the digital MAC increases the computation time as several consecutive digital operations will be needed to perform large VMM. (b) Near-memory computing architecture (NMC) is shown in this part and, in addition to the main processing unit, near memory processing units (NMPU) have been placed in vicinity of DRAM and NVM blocks. This reduces the data movement cost significantly as the commute distance of data is reduced by placing the processing unit close to the memory. Although in NMC the distance of memory to processing unit is decreased, there is still a significant amount of data commute in between for input, weight and output data. Also, the problem of high computation time due to the digital MAC exists. (c) In-memory computing architectures (IMC) implement computing within the memory. Specifically, in RS-based IMC, the RS-array can implement highly parallel VMM operation in one step and it also stored the weight matrix which will completely omit the weight movement during the operations. The only data movement in the IMC corresponds to input data. In-memory VMM (iVMM) is implemented over RS-based array in fully parallel manner by implementing several parallel in-memory MAC (iMAC) operations.

limited movements (i.e. data re-use). A more detailed analysis of this case needs to be considered for assessing the overall performances of RS-based VMM and system level analysis should address this question.

4.2 Current system-level propositions for RS-based VMM engines

In addition to physically implemented RS-based VMM engines, there are promising system-level propositions that are considering more complex ADC optimization and shared circuitry which could be viable for designing very energy efficient APUs. APUs are specialized hardware with a better performance in comparison with CPUs and GPUs to carry out specific tasks and applications. As can be seen in Figure 11(a), RS-based APUs are categorized

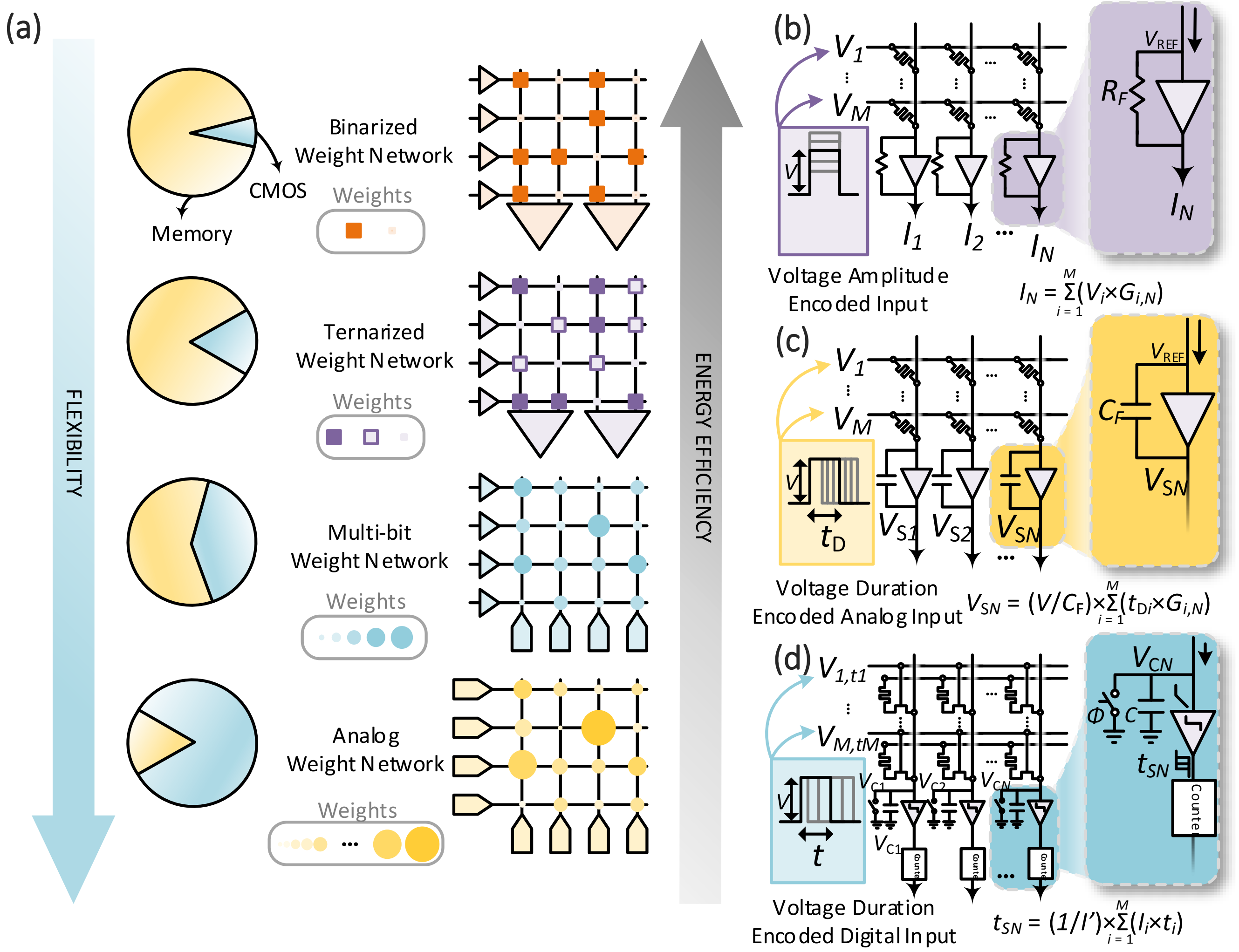

Figure 11. (a) Different RS-based APUs have been compared in terms of energy efficiency, storage efficiency, and flexibility for the particular case of the VMM operation. Each implementation presents a balance between memory functionalities (from binary to analog) and CMOS circuits overhead complexity and cost. (b) VMM engine design[88] with 0T1R analog weight network is illustrated based on input amplitude encoding and its corresponding sensing circuit with a feedback resistor in an op-amp follower block in the bit-line. (c) VMM engine design[95] with 0T1R analog weight network by utilizing input pulse duration encoding scheme and its corresponding sensing circuit for amplitude encoded analog output is depicted. (d) VMM engine design concept[96] for 1T1R weight network by using digital input pulse duration encoding scheme and its corresponding sensing circuit for pulse duration encoded digital output.

based on the precision of their weight cells into binarized, ternary, multi-level, and analog weight networks. The possibility of implementing wider ranges of applications with high resolution weight networks bring more flexibility in comparison with lower precision peers e.g. binarized and ternarized weight networks. On the other hand, low resolution APUs provides better energy efficiency and lower CMOS circuitry overhead which results in higher storage efficiency. One of the notable RS-based systems is ISAAC, which is a convolutional neural network accelerator[97]. ISAAC consists of tiles, which include eDRAM buffer, pooling unit, adders, and in-situ multiply accumulate (IMA) units. Inputs are sent through the eDRAM to IMA units which consist of RRAM crossbars and peripheral circuits (e.g. DAC and ADC) in an H-tree network topology. The dot-product computation of each crossbar is stored in the local sample and hold block. Subsequently, the 8-bit ADCs and shift-and-add circuits are carrying out the digitized outputs computations. This platform applied 16-bit input by digitizing it into 16 cycles of 1-bit pulse generated with 1-bit DACs. Also, 16-bit weights are distributed in eight columns with each RRAM cell providing 2-bit precision. Further enhancement of ISAAC has been proposed as NEWTON[98], which utilized various ADC optimization techniques such as adaptive ADC scheme and different multiplication methods (e.g. Karatsuba[99] and Strassen's algorithm[100]). This approach reduces the ADC computational overhead and leverage analog resolution of the MAC operation. In addition to these, NEWTON proposed buffer management techniques and a new mapping scheme to overcome the data communication and storage problems, respectively. The other important system to be noted here is PRIME[101], which is a general platform enabling both memory and computation modes by deploying three RS-based sub-arrays as its memory bank: memory sub-array, FF sub-array and buffer sub-array. The FF sub-array is utilized for both storage and computation purposes, memory array is employed only for storage purpose and buffer sub-array is used as the data buffer for FF sub-array. These three sub-arrays have been proposed as an optimization strategy for data trafficking. In terms of circuit overhead for RS-based VMM operation, PRIME avoids the need for high cost ADC circuits with reconfigurable precision (up to 8-bit) by designing a specific sense amplifier circuit block with a precision that is controlled with a counter. Since PRIME has been proposed as a ML-specific platform, rectified linear unit (ReLU) activation function and a block to support max pooling is added after the sense amplifier circuit to provide more efficient properties for applications like Convolution Neural Network (CNN). Alternatively, 3D-aCortex architecture[102] based on 3D NAND flash memories proposes to use time- domain encoding of the information that drastically reduces the cost of digital/analog conversions. In this strategy, both input and resulting output are consistently encoded into the pulse width enabling the pipelining of multiple VMM operations without converting data back into the digital domain. 3D-aCortex has been presented as a 3D integrated version of 2D-aCortex [103], which is a current-based architecture based on 2D NOR-flash memories and offers more than two orders of magnitude better area efficiency while maintaining the same throughput at the cost of low energy efficiency degradation in comparison with its 2D version. However, integrating the partial sums in the output for this time-domain design requires a large capacitor which is a bottleneck in terms of energy and area efficiency for a large sized VMM. To overcome this problem, the SIR VMM approach has been proposed in[104] based on the successive integration and rescaling (division) of the input bits. Unlike the previous time-domain encoding techniques, each bit of the digital input is encoded into binary pulses. To reduce the size of the load capacitor, in addition to this successive scheme, the accumulated charges will be divided via charge sharing mechanism. Utilization of SIR approach on the same architecture of 2D-aCortex using 1T1R 4-bit cells provide approximately 2.5× higher energy and area efficiency in comparison with conventional VMM methods. Three different design concepts of VMM engines for analog/digital input are encoded by the amplitude, analog input encoded by pulse duration, and digital input encoded by duration is shown in Figure 11(b-d).

Box 2: Performance metrics discussion for ML accelerators

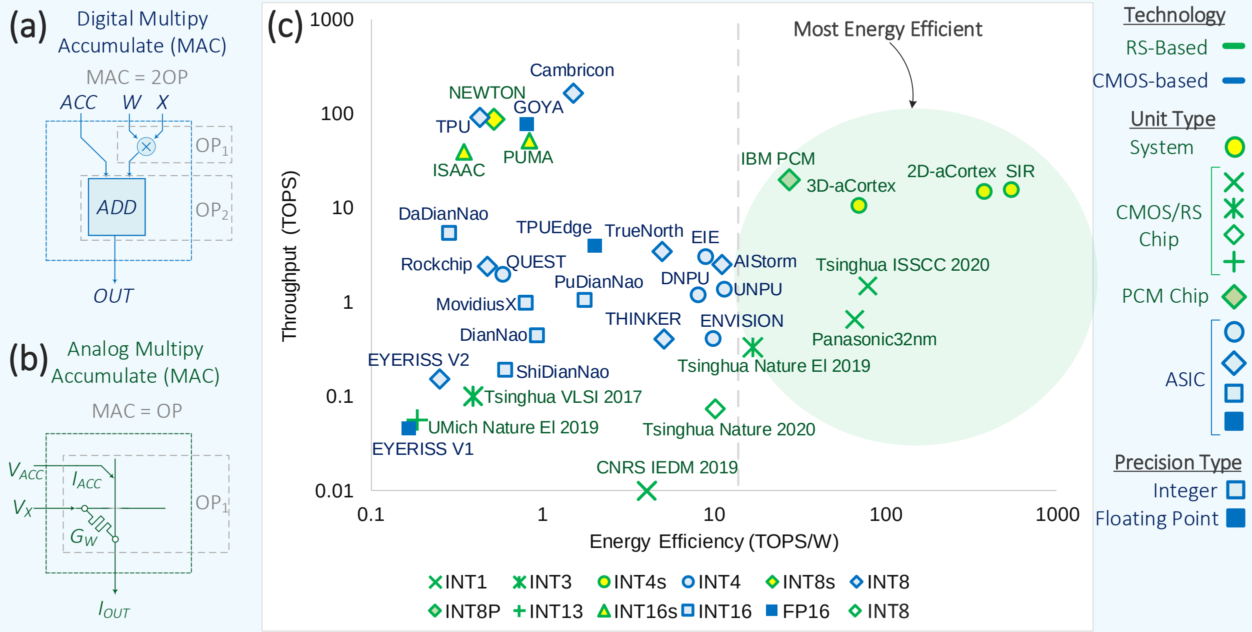

Figure 12. Implementation of multiply accumulate operation is depicted for both digital and analog domains. Also, the AI accelerator performance comparison is presented. (a) The digital implementation of MAC operation consists of two computational steps, multiplication and addition. Each of these steps are considered one operation (OP). Therefore, digital MAC is two OPs. (b) The analog implementation of MAC is shown on RS-based array by using Ohm's law and Kirchhoff's law in one computational step. The analog MAC unlike digital MAC is one OP. (c) The inference accelerator performances have been compared in terms of throughput and energy efficiency. The conventional CMOS-based digital ASIC chips, system solutions and RS-based chips are compared by considering the computation precision. Some of these systems or chips are reporting different performance numbers for multiple computation precision while here we demonstrate their performance for one of their reported precisions. Also, this plot may not be a full picture to show these chips and systems performance. As an example, although Eyeriss chips V1 [105] and V2[106] are showing a low energy efficiency below 0.5 TOPS/W in comparison with other systems, but they are very low power e.g. Eyriss V1 spends only around 1.67 pJ per MAC operation. This plot shows RS-based systems and chips are the most energy efficient ones. NEWTON[98], PUMA[107] and ISAAC[97] are also show promising throughput performance in comparison with state of art CMOS-based ASIC chips. However, these works did not consider the realistic device-level issues and integration challenges and their findings are only supported by simulation results (not experimentally implemented). As it can be seen, system solutions in most energy efficient region like aCortex systems and SIR has lower output precision in comparison with higher resolution peers like NEWTON, ISAAC and PUMA. Higher precision system solutions require more complex architecture and higher power consumption peripheral circuits which result in less energy efficiency, higher accuracy and higher flexibility in comparison with low precision systems.

Evaluation of the AI accelerators performance for training and inference in ML is a key step in todays competitive race toward building future AI platforms. Performance can be measured for various aspects like Inference Accuracy (IA), Storage Efficiency (SE), Energy Efficiency (EE), and Computational Efficiency (CE). Specific applications will favor some performance metric to another depending on the application constraint (e.g. embedded, high precision computing, low power, ...). Specialized hardware developed for ML applications are considering various precision, from 32-bit floating point to binary that makes consistent comparison of IA challenging. As a rule of thumb, a conventional ML algorithm can be implemented with limited accuracy of 8-bit integer without compromising too much inference performance. Lower accuracy requires the algorithms to be adapted significantly and becomes, consequently, more specialized to a specific application. For CE, the important metrics is the throughput, which defines the number of trainings/inferences that can be carried out by the training/inference engine in a certain amount of time. Conventionally, the numerical computing performance of digital computing systems is measured in Floating Point Operations Per Second (FLOPS). However, due to IA inhomogeneity, throughput unit is usually considered as Terra Operations Per Second (TOPS or TOP/s) for ML accelerators. Also, evaluation of the hardware throughput performance accounting for integration efficiency considers TOPS/mm^2. Regarding EE, the number of inference operations is normalized by energy consumption and results in TOPS/W (TOP/s/W or TOP/J). Finally, storage efficiency tracks the on-chip memory capacity for weights per unit area and is defined in MB/mm^2. In addition to TOPS, the term TMACS (Tera Multiply ACcumulates per Second) is widely used for defining the throughput of the digital neural network (NN) processors that are mostly focused on convolution-centric applications. In the digital APUs inference accelerator as depicted in Figure. 12(a), the Multiply Accumulate (MAC) operation consists of successive multiplication and addition operations. This means that, when accelerator manufacturers report the performance of their accelerator in TMACS, this value is equal to 2 times the performance in TOPS. While in analog VMM engines the MAC is considered as one operation (Figure 12(b)) which is a simple summation of currents over each synaptic device in the bit-line. In Figure 12(c), the performance comparison of the state of art inference accelerators have been shown based on throughput and energy efficiency metrics. In this comparison figure, we mostly selected the inference accelerators and tried to include different blend of designs including system solutions: ISAAC[97]NEWTON[98], 2D-aCortex[103], 3D-aCortex[102], SIR[104], PUMA[107], CMOS-based application specific integrated circuits (ASICs): ENVISION[108], AIStorm[109], DNPU[110], UNPU[111], EIE[112], TrueNorth[113], THINKER[114], EdgeTPU[115], TPU[116], Cambricon[117], GOYA[118], QUEST[119], PuDianNao[120], MovidiusX[121], DianNao[122], ShiDianNao[123], DaDianNao[124], RockChip[125], EYERISS V1[105], EYERISS V2[106], and fully integrated CMOS-RRAM VMM chips: UMich[76], Panasonic[93] , Tsinghua chips[54,77,89,92], CNRS[90],IBM PCM[57] by considering their peak performance values. Despite the common usage of the TOPS for ML processing unit evaluation, there is a concern as to whether this figure is not sufficiently comprehensive for direct evaluation with respect to a given application. For instance, embedded applications do not need necessarily to maximize throughput, but would require more drastic limitation in energy consumption. In addition, these reported performance numbers depend on various factors such as network compatibility with the computing platform. For example, different deep NN with the same number of MACs may result in a different throughput performance number on the same computing platform.

5 Conclusions and Perspectives

The competition toward an ideal VMM engine with high performance metrics is an ongoing race between research groups and companies these days. However, lots of factors need to be considered to achieve high reported performance numbers for each of these examples of hardware. In order to reach the reported performance numbers, overcoming the common problems results in reducing the throughput of the deep network inference is primary. Memory access is a limiting factor for achieving high processing speed for the processor as it is going to dominate the computation latency. Increasing the memory bandwidth, reducing the number of memory access in the DNN implementation by scheduling the computation steps, and increasing the arithmetic intensity of the layers, which defines the ratio of the computation over the memory access, are some of the possible solutions to reduce this effect on the accelerator throughput. To further tighten the gap of the tested throughput with the reported amount, there are some other strategies that needs to be mentioned such as: maximizing the parallelism to benefit from the full capacity of the hardware resources; reducing the input data transfer time; considering cooling and thermal envelop factor; and, the heterogeneous structure of today's processors. The approaches described above in this manuscript are examples of generic VMM engines that could be embedded within a digital platform. In order to sustain performances improvement, future hardware deployment based on the basic VMM operation should consider a more specialized VMM engine designed for a specific application.

Since RS-based VMMs are analog engines, a clear benefit would be to eliminate analog/digital conversions. There are numerous analog applications that could benefit from a local pre-processing of signals based on VMM operation. For instance, a RS-based VMM could be embedded into the front-end of sensors networks to compute directly analog signals. Other very demanding applications in terms of VMM operation are ML algorithms. Both synaptic weights and neurons are intrinsically analog elements. By integrating the analog neuron models directly into hybrid CMOS/RS processors, these platforms could maintain ultra-low power consumption and take advantage of purely analog computing. Note that spiking neural networks (e.g. neuromorphic hardware) would benefit from the same scheme since implementing digitally bio-realistic spiking neurons can become very costly while analog approaches seem very efficient. The trade-off here is to favor performances to flexibility since neurons models need to be specified a priori.

As it was discussed, lowering the resolution of the weight network will result a higher energy efficiency and lower flexibility VMM engine. High resolution weight network VMM platforms are more vulnerable to the impacts of device-level non-idealities like device-to-device and cycle-to-cycle variations and etc. To mitigate the impact of device non-idealities, in addition to device engineering to improve the intrinsic characteristic of the resistive switching memory cells, there have been several circuit- and system-level solutions have been presented. Some of the main ones are: using differential weights, WRITE and READ-verify method (closed-loop tuning), applying a hybrid training (in-situ and ex-situ) to reduce the impact of the faulty cells and non-idealities on the network's performance. Also, other solutions e.g. allocating multiple RS cells to resolve the weight precision issues, mapping largest weights to variation/fault-free crossbars to minimize the errors in larger platforms with multiple crossbar arrays, assigning larger weights to the most significant bit (MSB) and smaller weights to the least significant bits (LSB), and distinguishing between critical and non-critical weights to reduce the impact of faulty cells. There are several other non-ideality issues like conductance state-drift, hard faults (stuck-ON and stuck-OFF) and etc. which impacted the performance of the network. Existing solutions to conductance state-drift problem include periodical weight reprogramming and feedback designs which have limitations with high computational overhead and limited long-term effectiveness. From the other side of the spectrum, VMM engine can also be adapted to pure digital operation. For instance, binarized neural networks are machine learning models implemented with simple digital activation function (i.e. neurons), binary input vectors and binary weights. They cannot be used to map all ML algorithms but have demonstrated high performances for tasks that can tolerate binarized data. Their physical implementation, along with RS-based VMM, are highly cost effective and do not suffer from limitations such as accuracy and digital/analog conversion. Implementation of the neuron function with CMOS is based on simple XOR majority gates and digital memory in 1T1R configuration for the weights. This approach is analogous to biological neural networks that are operating with low resolution synapses and digital action potentials. Including the time encoding strategy used in biological networks into binarized neural networks could lead to an interesting physical implementation of bio-inspired computing. This strategy could potentially reconcile energy efficiency and flexibility of biological computing system that are still the most inspiring objective for future hardware development.

Acknowledgments

This work is supported by NSERC HIDATA project and ERC-CoG IONOS N° de convention de subvention 773228. Authors gratefully acknowledge Dr. Mohammad Bavandpour from University of California Santa Barbara, USA, Dr. Jason Eshraghian and Mr. Justin M. Corell from University of Michigan, Ann Arbor, USA, for their fruitful discussion and instructive information. Also, authors acknowledge Dr. Daniele Ielmini from Politecnico di Milano, Italy, Dr. D. Querlioz from C2N-CNRS, Drs. M. Bocquet and J. M. Portal from University of Aix-Marseille, and Dr. Meng-Fan (Marvin) Chang from National Tsing Hua University, Taiwan, for their helpful feedback and sharing data.

References

1. Moore, G. E. Cramming more components onto integrated circuits. *Proc. IEEE* **86**, 82–85 (1998).

2. Horowitz, M. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 10–14 (IEEE, 2014).

3. Esmaeilzadeh, H., Blem, E., Amant, R. S., Sankaralingam, K. & Burger, D. Dark silicon and the end of multicore scaling. In *2011 38th Annual international symposium on computer architecture (ISCA)*, 365–376 (IEEE, 2011).

4. Von Neumann, J. First draft of a report on the edvac. *IEEE Annals Hist. Comput.* **15**, 27–75 (1993).

5. Keckler, S. W., Dally, W. J., Khailany, B., Garland, M. & Glasco, D. Gpus and the future of parallel computing. *IEEE micro* **31**, 7–17 (2011).

6. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521**, 436–444 (2015).

7. Mutlu, O., Ghose, S., Gómez-Luna, J. & Ausavarungnirun, R. Processing data where it makes sense: Enabling in-memory computation. *Microprocess. Microsystems* **67**, 28–41 (2019).

8. Sze, V., Chen, Y.-H., Emer, J., Suleiman, A. & Zhang, Z. Hardware for machine learning: Challenges and opportunities. In *2017 IEEE Custom Integrated Circuits Conference (CICC)*, 1–8 (IEEE, 2017).

9. Ielmini, D. & Wong, H.-S. P. In-memory computing with resistive switching devices. *Nat. Electron.* **1**, 333–343 (2018).

10. Le Gallo, M. *et al.* Mixed-precision in-memory computing. *Nat. Electron.* **1**, 246 (2018).

11. Strukov, D., Indiveri, G., Grollier, J. & Fusi, S. Building brain-inspired computing (2019).

12. Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R. & Eleftheriou, E. Memory devices and applications for in-memory computing. *Nat. Nanotechnol.* 1–16 (2020).

13. Haj-Ali, A., Ben-Hur, R., Wald, N., Ronen, R. & Kvatinsky, S. Imaging: In-memory algorithms for image processing. *IEEE Transactions on Circuits Syst. I: Regul. Pap.* **65**, 4258–4271 (2018).

14. Li, C. *et al.* Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* **1**, 52 (2018).

15. Liu, S., Wang, Y., Fardad, M. & Varshney, P. K. A memristor-based optimization framework for artificial intelligence applications. *IEEE Circuits Syst. Mag.* **18**, 29–44 (2018).

16. Bojnordi, M. N. & Ipek, E. Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 1–13 (IEEE, 2016).

17. Mahmoodi, M., Prezioso, M. & Strukov, D. Versatile stochastic dot product circuits based on nonvolatile memories for high performance neurocomputing and neurooptimization. *Nat. communications* **10**, 1–10 (2019).

18. Cai, F. *et al.* Harnessing intrinsic noise in memristor hopfield neural networks for combinatorial optimization. *arXiv preprint arXiv:1903.11194* (2019).

19. Shin, J. H., Jeong, Y. J., Zidan, M. A., Wang, Q. & Lu, W. D. Hardware acceleration of simulated annealing of spin glass by rram crossbar array. In *2018 IEEE International Electron Devices Meeting (IEDM)*, 3–3 (IEEE, 2018).

20. Seo, J.-s. *et al.* On-chip sparse learning acceleration with cmos and resistive synaptic devices. *IEEE Transactions on Nanotechnol.* **14**, 969–979 (2015).

21. Sheridan, P. M. *et al.* Sparse coding with memristor networks. *Nat. nanotechnology* **12**, 784 (2017).

22. Kavehei, O. *et al.* An associative capacitive network based on nanoscale complementary resistive switches for memory-intensive computing. *Nanoscale* **5**, 5119–5128 (2013).

23. Eryilmaz, S. B. *et al.* Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array. *Front. neuroscience* **8**, 205 (2014).

24. Hu, S. *et al.* Associative memory realized by a reconfigurable memristive hopfield neural network. *Nat. communications* **6**, 1–8 (2015).

25. Wu, T. F. *et al.* Brain-inspired computing exploiting carbon nanotube fets and resistive ram: Hyperdimensional computing case study. In *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, 492–494 (IEEE, 2018).

26. Rahimi, A. *et al.* High-dimensional computing as a nanoscalable paradigm. *IEEE Transactions on Circuits Syst. I: Regul. Pap.* **64**, 2508–2521 (2017).

27. Eleftheriou, E. *et al.* Deep learning acceleration based on in-memory computing. *IBM J. Res. Dev.* **63**, 7–1 (2019).

28. Song, L., Qian, X., Li, H. & Chen, Y. Pipelayer: A pipelined reram-based accelerator for deep learning. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 541–552 (IEEE, 2017).

29. Sebastian, A. *et al.* Computational memory-based inference and training of deep neural networks. In *2019 Symposium on VLSI Technology*, T168–T169 (IEEE, 2019).

30. Gokmen, T. & Vlasov, Y. Acceleration of deep neural network training with resistive cross-point devices: Design considerations. *Front. neuroscience* **10**, 333 (2016).

31. Mahmoodi, M. *et al.* Ultra-low power physical unclonable function with nonlinear fixed-resistance crossbar circuits. In *2019 IEEE International Electron Devices Meeting (IEDM)*, 30–1 (IEEE, 2019).

32. Jiang, H. *et al.* A provable key destruction scheme based on memristive crossbar arrays. *Nat. Electron.* **1**, 548–554 (2018).

33. Nili, H. *et al.* Hardware-intrinsic security primitives enabled by analogue state and nonlinear conductance variations in integrated memristors. *Nat. Electron.* **1**, 197–202 (2018).

34. Gao, L., Chen, P.-Y., Liu, R. & Yu, S. Physical unclonable function exploiting sneak paths in resistive cross-point array. *IEEE Transactions on Electron Devices* **63**, 3109–3115 (2016).

35. Choi, S., Sheridan, P. & Lu, W. D. Data clustering using memristor networks. *Sci. reports* **5**, 10492 (2015).

36. Choi, S., Shin, J. H., Lee, J., Sheridan, P. & Lu, W. D. Experimental demonstration of feature extraction and dimensionality reduction using memristor networks. *Nano letters* **17**, 3113–3118 (2017).

37. Rahimi Azghadi, M. *et al.* Cmos and memristive hardware for neuromorphic computing. *Adv. Intell. Syst.* (2020).

38. Yan, B. *et al.* Rram-based spiking nonvolatile computing-in-memory processing engine with precision-configurable in situ nonlinear activation. In *2019 Symposium on VLSI Technology*, T86–T87 (IEEE, 2019).

39. Serb, A. *et al.* Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses. *Nat. communications* **7**, 1–9 (2016).

40. Gupta, I. *et al.* Real-time encoding and compression of neuronal spikes by metal-oxide memristors. *Nat. communications* **7**, 1–9 (2016).

41. Prezioso, M., Bayat, F. M., Hoskins, B., Likharev, K. & Strukov, D. Self-adaptive spike-time-dependent plasticity of metal-oxide memristors. *Sci. reports* **6**, 1–6 (2016).

42. Sun, Z. *et al.* Solving matrix equations in one step with cross-point resistive arrays. *Proc. Natl. Acad. Sci.* **116**, 4123–4128 (2019).

43. Zidan, M. A. *et al.* A general memristor-based partial differential equation solver. *Nat. Electron.* **1**, 411–420 (2018).

44. Moon, J. *et al.* Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nat. Electron.* **2**, 480–487 (2019).

45. Du, C. *et al.* Reservoir computing using dynamic memristors for temporal information processing. *Nat. communications* **8**, 2204 (2017).

46. Midya, R. *et al.* Reservoir computing using diffusive memristors. *Adv. Intell. Syst.* **1**, 1900084 (2019).

47. Chen, Y. Reram: History, status, and future. *IEEE Transactions on Electron Devices* **67**, 1420–1433 (2020).

48. Wu, T. F. *et al.* 14.3 a 43pj/cycle non-volatile microcontroller with 4.7 μs shutdown/wake-up integrating 2.3-bit/cell resistive ram and resilience techniques. In *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*, 226–228 (IEEE, 2019).

49. Chua, L. Memristor-the missing circuit element. *IEEE Transactions on circuit theory* **18**, 507–519 (1971).

50. Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. The missing memristor found. *nature* **453**, 80–83 (2008).

51. Xia, Q. & Yang, J. J. Memristive crossbar arrays for brain-inspired computing. *Nat. materials* **18**, 309–323 (2019).

52. Alibart, F., Gao, L., Hoskins, B. D. & Strukov, D. B. High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. *Nanotechnology* **23**, 075201 (2012).

53. Kim, H., Nili, H., Mahmoodi, M. & Strukov, D. 4k-memristor analog-grade passive crossbar circuit. *arXiv preprint arXiv:1906.12045* (2019).

54. Yao, P. *et al.* Fully hardware-implemented memristor convolutional neural network. *Nature* **577**, 641–646 (2020).

55. Yao, P. *et al.* Face classification using electronic synapses. *Nat. communications* **8**, 1–8 (2017).

56. Li, C. *et al.* Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. *Nat. communications* **9**, 1–8 (2018).

57. Ambrogio, S. *et al.* Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* **558**, 60–67 (2018).

58. Sassine, G. *et al.* Interfacial versus filamentary resistive switching in tio2 and hfo2 devices. *J. Vac. Sci. & Technol. B, Nanotechnol. Microelectron. Materials, Process. Meas. Phenom.* **34**, 012202 (2016).

59. Govoreanu, B. *et al.* 10× 10nm 2 hf/hfo x crossbar resistive ram with excellent performance, reliability and low-energy operation. In *2011 International Electron Devices Meeting*, 31–6 (IEEE, 2011).

60. Pi, S. *et al.* Memristor crossbar arrays with 6-nm half-pitch and 2-nm critical dimension. *Nat. nanotechnology* **14**, 35–39 (2019).

61. Lin, P. *et al.* Three-dimensional memristor circuits as complex neural networks. *Nat. Electron.* **3**, 225–232 (2020).

62. Adam, G. C. *et al.* 3-d memristor crossbars for analog and neuromorphic computing applications. *IEEE Transactions on Electron Devices* **64**, 312–318 (2016).

63. Burr, G. W. *et al.* Access devices for 3d crosspoint memory. *J. Vac. Sci. & Technol. B, Nanotechnol. Microelectron. Materials, Process. Meas. Phenom.* **32**, 040802 (2014).

64. Wang, C. *et al.* Cross-point resistive memory: Nonideal properties and solutions. *ACM Transactions on Des. Autom. Electron. Syst. (TODAES)* **24**, 1–37 (2019).

65. Adam, G. C., Khiat, A. & Prodromakis, T. Challenges hindering memristive neuromorphic hardware from going mainstream. *Nat. communications* **9**, 1–4 (2018).

66. Sung, C., Hwang, H. & Yoo, I. K. Perspective: A review on memristive hardware for neuromorphic computation. *J. Appl. Phys.* **124**, 151903 (2018).

67. Ambrogio, S. *et al.* Statistical fluctuations in hfox resistive-switching memory: Part ii—random telegraph noise. *IEEE Transactions on Electron Devices* **61**, 2920–2927 (2014).

68. Fantini, A. *et al.* Intrinsic switching variability in hfo 2 rram. In *2013 5th IEEE International Memory Workshop*, 30–33 (IEEE, 2013).

69. Alibart, F., Zamanidoost, E. & Strukov, D. B. Pattern classification by memristive crossbar circuits using ex situ and in situ training. *Nat. communications* **4**, 1–7 (2013).

70. Pan, W.-Q. *et al.* Strategies to improve the accuracy of memristor-based convolutional neural networks. *IEEE Transactions on Electron Devices* **67**, 895–901 (2020).

71. Chen, P.-Y. *et al.* Mitigating effects of non-ideal synaptic device characteristics for on-chip learning. In *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 194–199 (IEEE, 2015).

72. Hu, M. *et al.* Memristor-based analog computation and neural network classification with a dot product engine. *Adv. Mater.* **30**, 1705914 (2018).

73. Fantini, A. *et al.* Intrinsic program instability in hfo2 rram and consequences on program algorithms. In *2015 IEEE International Electron Devices Meeting (IEDM)*, 7.5.1–7.5.4 (2015).

74. Lee, H. *et al.* Evidence and solution of over-reset problem for hfo x based resistive memory with sub-ns switching speed and high endurance. In *2010 International Electron Devices Meeting*, 19–7 (IEEE, 2010).

75. Xia, L., Liu, M., Ning, X., Chakrabarty, K. & Wang, Y. Fault-tolerant training enabled by on-line fault detection for rram-based neural computing systems. *IEEE Transactions on Comput. Des. Integr. Circuits Syst.* **38**, 1611–1624 (2018).

76. Cai, F. *et al.* A fully integrated reprogrammable memristor–cmos system for efficient multiply–accumulate operations. *Nat. Electron.* **2**, 290–299 (2019).

77. Chen, W.-H. *et al.* Cmos-integrated memristive non-volatile computing-in-memory for ai edge processors. *Nat. Electron.* **2**, 420–428 (2019).

78. Dozortsev, A., Goldshtein, I. & Kvatinsky, S. Analysis of the row grounding technique in a memristor-based crossbar array. *Int. J. Circuit Theory Appl.* **46**, 122–137 (2018).

79. Carusone, T. C., Johns, D. A. & Martin, K. W. Analog integrated circuit design [m]. john wiley&sons (2011).

80. Hastings, A. *The an of analog layout* (Prentice hall New Jersey, 2001).

81. Ohnhäuser, F. *Analog-digital converters for industrial applications including an introduction to digital-analog converters* (Springer, 2015).

82. Murmann, B. *ADC Performance Survey 1997-2020* (2020).

83. Schreier, R., Temes, G. C. *et al.* *Understanding delta-sigma data converters*, vol. 74 (IEEE press Piscataway, NJ, 2005).

84. Walden, R. H. Analog-to-digital converter survey and analysis. *IEEE J. on selected areas communications* **17**, 539–550 (1999).

85. Harpe, P., Gao, H., van Dommele, R., Cantatore, E. & van Roermund, A. H. A 0.20 $text\{mm\}\hat{}23$ nw signal acquisition ic for miniature sensor nodes in 65 nm cmos. *IEEE J. Solid-State Circuits* **51**, 240–248 (2015).

86. Rabuske, T. & Fernandes, J. *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications* (Springer, 2017).

87. Hashemi, S., Anthony, N., Tann, H., Bahar, R. I. & Reda, S. Understanding the impact of precision quantization on the accuracy and energy of neural networks. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 1474–1479 (IEEE, 2017).

88. Bayat, F. M. *et al.* Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. *Nat. communications* **9**, 1–7 (2018).

89. Su, F. *et al.* A 462gops/j rram-based nonvolatile intelligent processor for energy harvesting ioe system featuring nonvolatile logics and processing-in-memory. In *2017 Symposium on VLSI Technology*, T260–T261 (IEEE, 2017).

90. Bocquet, M. *et al.* In-memory and error-immune differential rram implementation of binarized deep neural networks. In *2018 IEEE International Electron Devices Meeting (IEDM)*, 20–6 (IEEE, 2018).

91. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R. & Bengio, Y. Binarized neural networks. In *Advances in neural information processing systems*, 4107–4115 (2016).

92. Liu, Q. *et al.* 33.2 a fully integrated analog reram based 78.4 tops/w compute-in-memory chip with fully parallel mac computing. In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, 500–502 (IEEE, 2020).

93. Mochida, R. *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 *2018 IEEE Symposium on VLSI Technology*, 175–176 (IEEE, 2018).

94. Xue, C.-X. *et al.* 24.1 a 1mb multibit reram computing-in-memory macro with 14.6 ns parallel mac computing time for cnn based ai edge processors. In *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*, 388–390 (IEEE, 2019).

95. Marinella, M. J. *et al.* Multiscale co-design analysis of energy, latency, area, and accuracy of a reram analog neural training accelerator. *IEEE J. on Emerg. Sel. Top. Circuits Syst.* **8**, 86–101 (2018).

96. Bavandpour, M., Mahmoodi, M. R. & Strukov, D. B. Energy-efficient time-domain vector-by-matrix multiplier for neurocomputing and beyond. *IEEE Transactions on Circuits Syst. II: Express Briefs* **66**, 1512–1516 (2019).

97. Shafiee, A. *et al.* Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. *ACM SIGARCH Comput. Archit. News* **44**, 14–26 (2016).

98. Nag, A. *et al.* Newton: Gravitating towards the physical limits of crossbar acceleration. *IEEE Micro* **38**, 41–49 (2018).

99. Karatsuba, A. A. & Ofman, Y. P. Multiplication of many-digital numbers by automatic computers. In *Doklady Akademii Nauk*, vol. 145, 293–294 (Russian Academy of Sciences, 1962).

100. Huss-Lederman, S., Jacobson, E. M., Johnson, J. R., Tsao, A. & Turnbull, T. Strassen's algorithm for matrix multiplication: Modeling, analysis, and implementation. In *Proceedings of Supercomputing*, vol. 96, 9–6 (Citeseer, 1996).

101. Chi, P. *et al.* Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory. *ACM SIGARCH Comput. Archit. News* **44**, 27–39 (2016).

102. Bavandpour, M., Sahay, S., Mahmoodi, M. R. & Strukov, D. B. 3d-acortex: An ultra-compact energy-efficient neurocomputing platform based on commercial 3d-nand flash memories. *arXiv preprint arXiv:1908.02472* (2019).

103. Bavandpour, M., Mahmoodi, M. R. & Strukov, D. acortex: An energy-efficient multi-purpose mixed-signal inference accelerator. *accepted, IEEE J. Explor. Solid-State Comput. Devices Circuits* (2020).

104. Bavandpour, M., Sahay, S., Mahmoodi, M. R. & Strukov, D. Efficient mixed-signal neurocomputing via successive integration and rescaling. *IEEE Transactions on Very Large Scale Integration (VLSI) Syst.* (2019).

105. Chen, Y.-H., Krishna, T., Emer, J. S. & Sze, V. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE journal solid-state circuits* **52**, 127–138 (2016).

106. Chen, Y.-H., Yang, T.-J., Emer, J. & Sze, V. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE J. on Emerg. Sel. Top. Circuits Syst.* **9**, 292–308 (2019).

107. Ankit, A. *et al.* Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 715–731 (2019).

108. Moons, B., Uytterhoeven, R., Dehaene, W. & Verhelst, M. 14.5 envision: A 0.26-to-10tops/w subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm fdsoi. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 246–247 (IEEE, 2017).

109. Merritt, R. *Startup Accelerates AI at the Sensor* (2019).

110. Shin, D., Lee, J., Lee, J. & Yoo, H.-J. 14.2 dnpu: An 8.1 tops/w reconfigurable cnn-rnn processor for general-purpose deep neural networks. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 240–241 (IEEE, 2017).

111. Lee, J. *et al.* Unpu: An energy-efficient deep neural network accelerator with fully variable weight bit precision. *IEEE J. Solid-State Circuits* **54**, 173–185 (2018).

112. Han, S. *et al.* Eie: efficient inference engine on compressed deep neural network. *ACM SIGARCH Comput. Archit. News* **44**, 243–254 (2016).

113. DeBole, M. V. *et al.* Truenorth: Accelerating from zero to 64 million neurons in 10 years. *Computer* **52**, 20–29 (2019).

114. Yin, S. *et al.* A high energy efficient reconfigurable hybrid neural network processor for deep learning applications. *IEEE J. Solid-State Circuits* **53**, 968–982 (2017).

115. Google. *Edge TPU: Google's purpose-built ASIC designed to run inference at the edge* (2019).

116. Jouppi, N. P. *et al.* In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 1–12 (2017).

117. Cutress, I. Cambricon, maker of hauwei's kirin npu ip, build a big ai chip and pcie card (2018).

118. Armasu, L. *Move Over GPUs: Startup's Chip Claims to Do Deep Learning Inference Better* (2018).

119. Ueyoshi, K. *et al.* Quest: A 7.49 tops multi-purpose log-quantized dnn inference engine stacked on 96mb 3d sram using inductive-coupling technology in 40nm cmos. In *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, 216–218 (IEEE, 2018).

120. Liu, D. *et al.* Pudiannao: A polyvalent machine learning accelerator. *ACM SIGARCH Comput. Archit. News* **43**, 369–381 (2015).

121. Hruska, J. *New Movidius Myriad X VPU Packs a Custom Neural Compute Engine* (2017).

122. Chen, T. *et al.* Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. *ACM SIGARCH Comput. Archit. News* **42**, 269–284 (2014).

123. Du, Z. *et al.* Shidiannao: Shifting vision processing closer to the sensor. In *Proceedings of the 42nd Annual International Symposium on Computer Architecture*, 92–104 (2015).

124. Chen, Y. *et al.* Dadiannao: A machine-learning supercomputer. In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, 609–622 (IEEE, 2014).

125. Rockchip. *Rockchip Released Its First AI Processor RK3399Pro NPU Performance up to 2.4TOPs* (2018).