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

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

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, 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 colocating 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 low power. Theoretically, this directly translates to a major boost in energy efficiency and computational throughput, but various practical challenges remain. 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 machine learning (ML) inference, that of vector-matrix multiplication (VMM), is presented. The monolithic integration of RS memories with complementary metal–oxide–semiconductor (CMOS) integrated circuits is presented as the core underlying technology. The key existing design choices in terms of device-level physical implementation, circuit-level design, and system-level considerations is reviewed and an outlook for future directions is provided.

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 complementary metal–oxide–semiconductor (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\textsuperscript{[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 central processing units (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 the energy consumption perspective and the 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 graphics processing units (GPUs)\textsuperscript{[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 to offer important speed improvements. As MAC operation represents the most intensive calculation required for ML algorithm implementation, it explains why GPUs have led to important breakthroughs in the acceleration of ML by enabling training and operation of deep neural networks (NNs)\textsuperscript{[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 levels\textsuperscript{[7]} and 2) as in CPUs, 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 1a). 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.\textsuperscript{[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 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 colocating 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)\textsuperscript{[8]} and spin-transfer torque magnetic random access memory (STT-RAM).\textsuperscript{[9]} Beyond digital memory technologies, in-memory computing based on nonvolatile resistive switching (RS) devices monolithically integrated on CMOS is opening new perspectives for ultra-efficient MAC operation engine development.\textsuperscript{[10]} First, the 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.\textsuperscript{[10–12]} Second, in-memory computing represents a new physical implementation of the basic MAC operation with the potential for important improvements with respect to the same criterion.

In this article, we will review the main limitations and opportunities of in-memory computing with resistive memories for the MAC operation engine, also known as the vector-matrix multiplication engine (VMM engine). On this basis, as shown in Figure 1b, 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 the 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\textsuperscript{[13]} such as image processing,\textsuperscript{[14–15]} combinatorial optimization,\textsuperscript{[16–20]} sparse coding,\textsuperscript{[21,22]} associative memory,\textsuperscript{[23–27]} deep learning inference/training,\textsuperscript{[28–31]} unclonable functions\textsuperscript{[32–35]} principle component analysis,\textsuperscript{[36,37]} spiking NNs\textsuperscript{[38–42]} solving linear,\textsuperscript{[43]} and partial differential equations\textsuperscript{[44]} and reservoir computing.\textsuperscript{[45–47]} Our intent is to provide a comprehensive analysis to assess the novelty of the reviewed examples and discuss different design choices to better understand this emerging class of hardware and 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 NN structure (Figure 2a). 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 2b). The precision of 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 current $I$ is equal to voltage $V$ multiplied by conductance $G$ ($V \times G = I$) (Figure 2b,c). The advantages of this approach are twofold: 1) only a single time step is required to compute the multiplication versus multiple time steps in digital implementation and 2) energy consumption is considerably lower. Considering projected resistive memories performance for an average resistance of $R = 1 \text{M} \Omega$ with a read-voltage of 0.1 V with a pulse duration of $1 \text{ns}$, the energy consumption equals $E_1 = 0.1 \times 10^{-7} \times 10^{-9} = 10^{-17} \text{J}$. Note that with today’s performances, the energy calculation should consider $R = 10 - 100 \text{k} \Omega$, $V = 0.1 \text{V}$, and $t = 1 \mu$s leading to
$E_2 = 0.1 \times 10^{-5}/6 \times 10^{-6} = 10^{-12}/13$ J. This energy consumption should be compared with 8-bit digital multiplication of $E_3 = 0.2$ 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 conducted 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 conducted within multiple parallel channels of the crossbar simultaneously in a single clock cycle with multiplications. For the sake of comparison, one 8-bit full-adder uses \approx 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 operations, multiplication and addition, 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 occur due to nonideal parameters such as physical constraints, circuit-level challenges, and system-level challenges. c) Manuscript organization.

2.2. Dot-Product Precision

RS devices have been developed following two main research directions. On the one hand, the RS mechanism has been
investigated as a potential solution for the development of universal memory. This kind of binary memory could combine high switching speed (sub-ns), low energy (pJ range), and high endurance (1012 cycles) of DRAM and SRAM with nonvolatility (>10 years retention) and scalability (<10 nm) (Figure 3a,b). In this article, we focus on a subclass of RS devices, known as redox-based random access memory (ReRAM), but the same conclusions can be applied to the other class of resistive memory such as FeRAM, PCRAM, and CBRAM. Various ReRAM cell candidates, among which HfO$_x$ and TaO$_x$ ReRAM are the best representatives (Figure 3c,d), are already integrated in the fabrication lines of industry and integrated with CMOS technology.[49] 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 multilevel cells. This kind of device can either implement a 1-bit dot-product or a low resolution, e.g., <3-bit dot-product.[50]

On the other hand, many research groups have focused on the RS mechanism for memristors or memristive device implementation (Figure 3d). The association between the theoretical concept proposed by Chua[51] and a possible physical implementation of this new circuit element[52] has opened new perspectives for circuit design and especially for VMM. In the ideal memristor framework, RS 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 analog switching using various switching mechanisms and materials such as transition metal oxides, ferroelectric tunnel junctions, or more exotic materials (see a previous study[53] for a review of the different options). Memristive devices have demonstrated analog switching controlled by the analog pulses of voltage equivalent to 8-bit accuracy, paving the way

---

Figure 2. a) A basic NN structure including the input vector, weight matrix, and output vector. b) Schematic of the digital and analog VMM 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 the 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 device conductance and the output are 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 BEOL process.
for 8-bit dot-product. The 8-bit accuracy has been demonstrated on discrete devices and only 4-bit–5-bit resolution has been reported for integrated devices due to parasitic effects induced from other circuit elements.

The maturity of memristive technologies is not as developed as the ReRAM technology, which results in inferior performance regarding endurance, retention, and speed. Importantly, memristive devices and ReRAM are both subject to additional drawbacks when cointegration with CMOS is considered. Reading the resistive state can be implemented with high-speed and low-voltage CMOS circuitry. Nevertheless, forming the resistive element and programming their resistive state with high speed (e.g., <10 ns) require large voltage (e.g., >2 V) that could prevent cointegration with ultrasmall pitch CMOS technologies. Using a thick-oxide transistor to tolerate this high voltage is another cointegration issue in these platforms that has to be considered.

There are still several research opportunities in this area and efforts have 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 NNs that rely deeply on the high-accuracy calculation of the synaptic weights during training. In that scope, innovations in integration schemes could greatly improve the accuracy of the memristor-based VMM. For instance, while ReRAMs differ from analog memristive devices by the difficulty to access intermediate resistance states, there is, in principle, no physical limitation to have multilevel analog states in ReRAM. HfO$_x$-based ReRAM, which usually exhibits sharp SET and semi-gradual RESET, can be better controlled using the analog current limitation mechanism through an access transistor to implement analog switching close to 5-bit precision. 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 a previous study 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 nonvolatile 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 with 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.

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 previous studies, paving the way to surpass the scaling limitations.
Box 1. Analysis of Nonideal Parameters of RS Memories That Impact Neural Network Accuracy

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

Figure 4. Schematic classification of a memristor-based system’s nonidealities according to the way they impact 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 hyperparameters 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. The last column “Training Strategies” contains the two main approaches to train a network with RS memories.

Designing an RS-based system compatible with the established microelectronic industrial technologies and large-scale production is only one part of the challenge. As RS devices have inherent physical imperfections, it is necessary to find efficient ways to deal with them. The impact level of such nonideal 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 networks (ANNs). 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 Mixed National Institute of Standards and Technology (MNIST) database corresponds to the proportion of the 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.

In the in situ scheme, the training is conducted 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 nonideal parameters that affect the conductance writing accuracy 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 being 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 makes it viable to apply advanced methods to mitigate nonideal parameters related to writing.

Finally, a hybrid strategy showed some interesting results by fine tuning the network weights after transfer. To better understand the different impacts of RS-based system nonideal 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 interdependence between the different parameters. For example, the switching endurance, which represents the average number of cycles before losing RS behavior, directly impacts minimum and maximum conductance values over cycles, which in turn contribute to determining the total number of conductance state. Therefore, poor switching endurance could indirectly lead to low number of conductance states or even failure such as stuck at fault where only one conductance state exists. The impossibility to update the conductance decreases the ANN accuracy 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 or the overall asymmetry of conductance variation. Further work should be conducted on the interactions between all nonideal parameters to clarify their direct and indirect impact on the accuracy of physically implemented ANN, which could help the design and demonstration of mitigation strategies.
of Flash and DRAM. In addition, the two-terminal structure of RS devices enables ultradense 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 the monolithic 3D integration in CMOS back end of line (BEOL). This ideal approach (see Figure 2d) 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 with 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 of stacking 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.[34,62] There are still important engineering challenges to address to bring these concepts to their full potential: 1) compatibility of advanced lithography steps with BEOL metal layer, 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 ultrafine 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 paths, currents coming from other resistive cells in the array, are preventing an accurate reading of each resistive element individually. ReRAM and memristive devices can be addressed with or without the use of a selector. On the one hand, ReRAM requirements have favored optimizations toward 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. The resulting integration scheme is then only considered as a pseudocrossbar array. Two-terminal selectors, such as threshold switching elements or nonlinear 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.[63] 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 have to be switched for each read operation. A detailed review in this topic can be found in a previous study.[64] 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 as 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). Note that this consideration is only true for read operation. Programming individual devices in passive crossbars still remains a challenge for very large arrays as half-biased elements during programming affect severely leakage currents and state disturbance. Nevertheless, more exploratory in-memory computing paradigms such as neuromorphic computing, or bioinspired SNNs (SNN), can also take advantage of a similar principle and could leverage programming inaccuracy with parallel programming schemes and low-accuracy requirements. 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 the practical integration of crossbars on chips still requires access transistors at the $N$ input lines and $M$ output columns of the crossbar, thus leading to $(N + M)(N \times M)$ actual footprint on silicon. There is, consequently, a strong interest in improving passive crossbar dimensions above the $64 \times 64$ size report so far.[55]

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 manipulations (adding multiple dot-products, 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 5a,c). Adding the infinite size of dot-products results in infinite time in the digital scheme and it results in infinite power for the Kirchhoff’s law-based approach.

Practically, memristor-based VMM has been reported for a matrix size of up to $128 \times 64$. While this was demonstrated with the pseudocrossbar 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 $64 \times 64$ VMM operation was demonstrated in a previous study[55] using a purely passive crossbar with a more advanced patterning process (<200 nm). Dot-product demonstrations with other integrated approaches[77,78] are today limited to small vector 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.[61] Alternatively, increasing the mean resistance of RS devices would increase scalability significantly by reducing power consumption at the cost of a 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 impedance, and capacitance. The inference delay is calculated based on the Elmore delay model as follows:

$$t_{inf} = t_{setting} + \sum_{i=1}^{4} \tau_i$$  \hspace{1cm} (1)
where $t_{setting}$ is the settling time of the output circuit. As shown in Figure 5b, the parameters $\tau_1$, $\tau_2$, $\tau_3$, and $\tau_4$ are the delays from a row, RS cell, column, and output circuit, respectively. By considering the low resistance state (LRS) resistance of the device much larger than the interconnect resistance between two adjacent cells, the delays $\tau_3$ and $\tau_4$ are dominant in very large arrays. By increasing the LRS of the RS cell, the inference time delay increases as it impacts both $\tau_3$ and $\tau_4$. 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 $\tau_4$ 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 pseudocrossbar scalability challenges can also be related to the computing performance (e.g., accuracy). Unlike digital approaches where input digital signal 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 memory cells and the one of metal interconnects becomes critical in large crossbar arrays (Figure 5d). The same bias applied to the word-line is seen differently by each cell in the crossbar due to linear voltage decreases that lead to a decrease in 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 conducted in one step. Note that small VMM dimensions are largely used for convolutions in convolutional neural networks (CNN). In conclusion, the scalability of memristor-based VMM operation represents a future research direction that requires innovative solutions at both technological and system levels.

3. Circuit Design Challenges for VMM Implementation

3.1. Background

As mentioned previously, the projected energy consumption for a single dot-product operation can indeed be as small as 0.01 fJ, whereas 0.2 pJ is consumed with the 8-bit digital VMM based on 45 nm CMOS technology node. However, this comparison is not a complete picture as it does not consider energy consumption for I/O signals generation. A more rigorous evaluation of
the memristor-based dot-product energy consumption should be done by considering the 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 ≈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 as $E_{DAC} + E_{ADC} = 2 \times 0.1 \times 10^{-3} = 0.2$ pJ. This simple example therefore highlights the importance of the overhead circuitry in the assessment of VMM engine performance. While most of the approaches so far have been using software emulated or custom on printed circuit boards (PCBs), there have been recently only a few fully integrated chip demonstrations. These demonstrations’ benefits are twofold: 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 define the application field of the VMM engine and impact 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, ML algorithms, combinatorial optimizations, or solving linear and partial differential equations. In these applications, the VMM operation has to be integrated into a digital environment used to manage the higher-order functions such as data management and VMM definition/programming. Generating an analog input voltage from digital input data can be implemented with DAC, which implies that digital circuits can be estimated by dynamic power of the analog circuits. The power dissipation of digital circuits can be estimated by

$$P_D = f_{2B} C_p V^2 + P_{Leakage}$$

(2)

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 the technology node (around several pico-Watts for an inverter in 65 nm technology from 1 V power supply). For resistive DAC the main analog power is from the OP-AMP follower output stage, which usually uses a class-A output stage that has a maximum power efficiency of 50%. So, the analog power can be estimated using

$$P_A = n \times V^2/R$$

(3)

![Figure 6](image-url)
where $n$ and $R$ parameters are the number of devices which have been driven by the DAC and minimum RS device resistance, respectively. Assuming that 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 7a. It has been shown that power consumption is almost proportionate with the number of devices below 100 MS s⁻¹ (mega sample per second) operating frequency and this is because the analog power dominates when the quantity of RS devices becomes relatively large. While in higher operating frequencies than 100 MS s⁻¹ the power consumption will be impacted mainly by the 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 is 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

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

(4)

where $W$ and $L$ are the width and length of the input pairs, respectively. Parameter $K$ is Boltzmann’s constant, $C_{\text{ox}}$ is the gate capacitance per unit area, and $f_1$ and $f_2$ are the low-corner and high-corner frequencies, respectively.[80] The matching of the resistor is described as follows

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

(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.[81] Figure 7b shows the estimated area of resistive DAC versus the operating frequency and the effective number of bits (ENOBs). The area changes almost linearly with the operating frequency and exponentially with the ENOBs. A similar approach can be used for estimating the area and power consumption for the DAC with a different architecture. In addition to the undesirable high energy consumption of the high-resolution DACs, delivering perfect analog input signals on each memory cell is challenging as it can be easily deteriorated by crossbar arrays imperfections. As mentioned previously, voltage drop along the metal lines (Figure 5c) 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 in a previous study.[73] 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 are encoded with the analog voltage amplitude signals is the nonlinearity 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 preprocessing, including the effect of RS devices’ nonideal parameters into the analog input, but can become quickly very complicated if the high variability in RS device is to be integrated in preprocessing. Alternatively, Cai et al.[77] proposed a method to solve this limitation by encoding

![Figure 7](image-url)
the analog input signal with pulse width modulation. This strategy comes at the cost of multiple clock cycles for each encoded input but mitigates I-V nonlinearity. 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 the 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 nonidealities 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 using analog inputs in their digitized form. Each bit from the analog input number is computed sequentially from the least significant bit (LSB) 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 an RS-based VMM operation are analog currents that have to be converted into digital numbers. A straightforward solution is to use ADC and transimpedance amplifiers (TIAs). 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. Using high-resolution ADC in such arrays is one option for distinguishing the analog output levels, which require 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), etc.\[82\]

In a typical RS-based VMM engine, the most important metrics to consider for the ADCs are their resolution, their sampling frequency, their power consumption, and DSM have slightly smaller form factors (Figure 8d) and significantly better SNDR. Voltage control oscillator (VCO)-based ADC or SAR ADC are more suitable to smaller technology node implementations as they do not rely on high-gain/bandwidth amplifiers that are limited by intrinsic transistor gain.\[87\] If the inference operation takes longer than 10 ns, low-resolution/high-speed-flash ADC can be applied via time multiplexing to minimize die area as an 8-bit ADC is usually needed for a typical NN to achieve more than 90% classification accuracy.\[77\]\[88\] 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 8b marked the lowest possible ADC power consumption for a given sampling frequency, and the dashed line shown in Figure 8c 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

\[ P = FOM_{W, min} \times 2^{\text{ENOB}} \times f_s \]

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

\[ \text{SNDR}_{\text{max}}(\text{dB}) = (\text{ENOB} \times 6.02) + 1.76 \]

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

\[ \text{SNDR}_{\text{max}}(\text{dB}) = 165 - 10 \log_{10}\frac{\text{ERBW}}{FOM_{W}} \]

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

\[ \text{Area (mm}^2) = A - (10^{-4} \times FOM_{W}) \]

where A is the technology-dependent factor that equals \( 2 \times 10^{-3} \) for 14 nm technology. The analysis shown earlier 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 a four-fold larger overall capacitance, as the thermal noise at the input of ADC equals \( KT/C \) (where K is the Boltzmann constant).

\[ FOM_{W}(f/\text{CONV.STEP}) = \frac{P}{2^{2\text{ENOB}} \times \min(f_s, 2\text{ERBW})} \]

where ENOB is the ENOBs. In general, the achievable best FOMs decreases along with the increase in frequency, e.g., doubling \( f_s \) or increasing 1-bit resolution postulates quadrupled power consumption (Figure 8e). In addition, reducing \( FOM_{W} \) demands an increase in die area, e.g., 50% power reduction or 1-bit more resolution have 25% more die-area (Figure 8f). 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 (SAR) (ADC) or delta-sigma (DSM) ADC can be utilized as (SAR) and DSM have smaller form factors (Figure 8a) and significantly better SNDR.

The achievable performance of the ADC can be predicted by two well-known figure of merits (FOMs)\[84-86\] as

\[ FOM_5(\text{dB}) = \text{SNDR} + 10 \log_{10}\left(\frac{\text{ERBW}}{P}\right) \]

where ERBW is the bandwidth of ADC and P is the total power dissipation.
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.\(^{[84]} \) Figure 8e shows the relationship between Shreier’s FOM and sampling frequency, the maximum achievable FOM at a low frequency (<10 MHz) is 192 dB, and at higher frequencies (>10 MHz), the best achievable FOM equals

\[
\text{FOM}_{S, \text{max}}(\text{dB}) = 192 - (10 \times \log_{10} \frac{F_{\text{BW}}}{10})
\]  

(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. 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.

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 have used discrete integrated circuit components connected to the RS array and they did not present a complete integrated system in a single chip. 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 shown in Figure 9, choices for input, output, and weight cell include binary, ternary, multi-bit, and analog. However, selecting a design choice is directly dependent on the target application requirements and its functional aspects, e.g., accuracy, speed, 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 nonvolatile intelligent processor (NIP) has been designed using 4 kb 1T1R binary HfO$_x$-based cells and 150 nm CMOS technology. This work proposes a nonvolatile 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 the input-controlled access transistor scheme in 1T1R array. The other physically implemented chip is a binary VMM engine shown in a previous study using 2T2R differential weights with the input-controlled access transistor scheme and a precharged sense amplifier (PCSA) circuit. This chip was developed for binarized NN demonstration but consists, essentially, of a binary dot-product operation. The 2 kb HfO$_x$-based RS devices have been integrated on top of the fourth metal layer in CMOS 130 nm technology node. The PCSA circuit is differential and connected to both bit-lines of the 2T2R cells in each column. Due to the binarized NN properties, 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 conducted by popcount gates. This chip is purely digital and it is free from any D/A or A/D conversion that results 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 the 65 nm CMOS technology node. This implementation proposed new circuit peripherals and architecture-level ideas 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 subarrays. Partial MAC results computed from each subarray are added together to compute a partial MAC. To avoid using costly DAC circuits, this work proposes the dual word line driver (D-WLDR) circuit to apply inputs in both memory and inference modes. These circuits include small digital buffers occupying small areas 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 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 generates 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 minimizes the offset in the 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 the 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 a limited array size (VMM is limited to dimension 12). In a previous study, a 158 kb VMM engine is designed in 130 nm CMOS technology and the issues 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 are attempted to be mitigated. A signed weight 2T2R cell has been used to reduce the column’s sensing current by getting benefit from the differential current. In this work, a quasi-3-bit weight (seven level) is used by positive and negative 1T1R cells that locally cancel 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 provides a high energy efficiency of 78.4 TOPS/W when sensing the output by 1-bit precision and a high inference accuracy around 94% for MLP of the MNIST classification task with 8-bit sensing precision in both ADC stages of the network. For multilevel weight design choices, Yao et al. 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 eight-element RS devices. Analog inputs are encoded into 8-bit binary sequential pulses in eight time intervals and applied via the external voltage generator to PE chips. Each PE chip includes four ADC blocks with 8-bit precision to sense 128 x 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 is connected via a pair of S/H blocks. In the 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 whereas the ADC conducts 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⁻¹) and it enhanced the inference accuracy for the MNIST classification task up to 95.57%. The first demonstration of the VMM engine with analog weight deploying a passive RS crossbar by the size of 54 x 108 monolithically integrated with CMOS in 180 nm technology node on a single chip is shown in a previous study. In this work, a charge-based inference is targeted to overcome the I–V nonlinearity 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. DAC then applies the corresponding 6-bit weight-modulated input pulse into the array. The bit-line-accumulated charges are sensed by an incremental charge-integrating ADC. The high-resolution hybrid 13-bit ADC circuit is placed in both rows and columns to enable bidirectional inference operation and it comprises a 5-bit first-order incremental ADC, an 8-bit SAR ADC, and an additional 1-bit redundancy stage. The OpenRISC processor with 64 kB SRAM along with timing generation blocks has 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 bidirectional inference capability make this platform highly flexible to implement different blends of ML applications. However, this flexibility adds cost as the number of ADCs is doubled besides the fact that high-resolution ADC consumes more power and area as well. Each of these VMM engine design choices offers different performance behaviors 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 shown in Table 1.

4. System-Level Development of RS-Based VMM Engines

4.1. Leveraging the Cost of Mixed Analog/Digital Approaches and Data Trafficing

Performances of VMM engines appear 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 circuit cost should evolve favorably...
by increasing the dimensions of RS-based VMM engines. Indeed, as \( N \times M \) DACs and ADCs are required to drive an \( N \times M \) crossbar array, 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 correspond 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 shown in a previous study,\(^{[8]} \) there are optimization strategies that could take advantage of input data reuse and moving weights during computation. This is of particular interest for CNNs as convolutions use input data multiple times during scanning. As shown in Figure 10a,b, in the conventional von Neumann computing systems and near-memory computing (NMC) architectures, all input, weight, and output data move between the processing unit and memory. However, the traveling distances in NMC systems are significantly smaller than the conventional von Neumann computing architectures (few millimeters in high-bandwidth memory integrated with interposer technology). On the other hand, in-memory computing of the MAC operation proposes to permanently store the matrix component into a dedicated nonvolatile memory, thus reducing drastically the data movement for these parameters (Figure 10c). Nevertheless, I/O data still have 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 CNN and could benefit from limited movements (i.e., data reuse). A more detailed analysis of this case has 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 better performance in comparison with CPUs and GPUs to conduct specific tasks and applications. As shown in Figure 11a, RS-based APUs are categorized based on the precision of their weight cells into binarized, ternary, multilevel, and analog weight networks. The possibility of implementing wider ranges of applications with high-resolution weight networks brings more flexibility in comparison with lower precision peers, e.g., binarized and ternarized weight networks.
other hand, low-resolution APUs provide better energy efficiency and lower CMOS circuitry overhead which result in a higher storage efficiency. One of the notable RS-based systems is ISAAC, which is a CNN accelerator. 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 ReRAM 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 S and H block. Subsequently, the 8-bit ADCs and shift-and-add circuits carry 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 ReRAM cell providing 2-bit precision. Further enhancement of ISAAC has been proposed as NEWTON, which utilized various ADC optimization techniques such as the adaptive ADC scheme and different multiplication methods (e.g., Karatsuba and Strassen’s algorithm). 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 data communication and storage problems, respectively. The other important system to be noted here is PRIME, which is a general platform enabling both memory and computation modes by deploying three RS-based subarrays as its memory bank: memory subarray, full function (FF) subarray, and buffer subarray. The FF subarray is utilized for both storage and computation purposes, memory array is used only for storage purpose, and buffer subarray is used as the data buffer for FF subarray. These three subarrays have been proposed as an optimization strategy for data trafficing. 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. As PRIME has been proposed as an ML-specific platform, rectified linear unit (ReLU) activation function and a block to support max pooling are added after the sense amplifier circuit to provide more efficient properties for applications like CNN. Alternatively, 3D aCortex architecture\cite{101} 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\cite{102}, which is a current-based architecture based on 2D NOR flash memories and offers more than two orders of magnitude and 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 was proposed in a previous study\cite{103} 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. The utilization of the SIR approach on the same architecture of 2D aCortex using 1T1R 4-bit cells provides \( \frac{tSN}{C2} \times 2.5 \) 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, that are shown in Figure 11b–d.

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 have to be considered to achieve high reported performance numbers for each of these examples of hardware. To reach the reported performance numbers, overcoming the common problems results in reducing the throughput of the deep network inference which is
### Box 2. Performance Metrics Discussion for ML Accelerators

| Technology         | Top Type   | Unit Type   | Precision Type |
|--------------------|------------|-------------|----------------|
|                   | System     | Chip        | Integer        |
|                   |            |             | Floating Point |
| RS-Based           |            |             |                |
| CMOS-based         |            |             |                |
| PUMA               |            |             |                |
| CMOS/RS Chip       |            |             |                |
| PCM Chip           |            |             |                |
| ASIC               |            |             |                |

#### Figure 12. Implementation of MAC is shown for both digital and analog domains. 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 is considered one operation (OP). Therefore, digital MAC is two OPs.
- **b)** The analog implementation of MAC is shown on the RS-based array 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 report different performance numbers for multiple computation precisions, whereas 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 Eyriis chips V1\(^{[140]}\) and V2\(^{[105]}\) show a low energy efficiency below 0.3 TOPS/W in comparison with other systems, but they are very low power, e.g., Eyriis V1 spends only around 1.67 pJ per MAC operation. This plot shows that RS-based systems and chips are the most energy-efficient ones. NEWTON\(^{[97]}\), PUMA\(^{[106]}\), and ISAAC\(^{[96]}\) also show promising throughput performance in comparison with state-of-the-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 shown, system solutions in most energy-efficient region like aCortex systems and SIR have a 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 today’s competitive race toward building future AI platforms. Performance can be measured for various aspects such as 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 precisions, from the 32-bit floating point to binary that makes the consistent comparison of IA challenging. As a rule of thumb, a conventional ML algorithm can be implemented with a limited accuracy of the 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 conducted 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, the 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 tera multiply accumulates per second (TMACS) is widely used for defining the throughput of the digital NN processors that are mostly focused on convolution-centric applications. In the digital APUs’ inference accelerator, as shown in Figure 12a, the MAC operation consists of successive multiplication and addition operations. This means that, when the accelerator manufacturers report the performance of their accelerator in TMACS, this value is equal to 2 times the performance in TOPS, whereas in analog VMM engines the MAC is considered as one operation (Figure 12b) which is a simple summation of currents over each synaptic device in the bit-line. In Figure 12c, the performance comparison of the state-of-the-art inference accelerators has been shown based on throughput and energy-efficiency metrics. In this comparison...
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 have to be mentioned such as maximizing parallelism to benefit from the full capacity of the hardware resources, reducing the input data transfer time, considering the cooling and thermal envelop factor, and, the heterogeneous structure of today’s processors. The approaches described earlier in this manuscript are examples of generic VMM engines that could be embedded within a digital platform. To sustain performance improvement, future hardware deployment based on the basic VMM operation should consider a more specialized VMM engine designed for a specific application.

As 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 preprocessing of signals based on VMM operation. For instance, an 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 operations 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 ultralow power consumption and take advantage of purely analog computing. Several different activation functions can be applied on the output circuits of the resistive memory arrays, e.g., ReLU, sigmoid, tangent hyperbolic \((\tanh)\), and \(\sin^2(x)\). Different CMOS circuit designs have been presented for various activation functions. Studying the impact of these functions on the network performance, designing analog neuron activation functions in CMOS and matching neuron operations with synaptic arrays represent a future challenge that will have to be carefully addressed by the research community. Note that SNNs (e.g., neuromorphic hardware) would benefit from the same scheme as implementing digitally biorealistic spiking neurons can become very costly whereas analog approaches seem very efficient. The trade-off here is to favor performances to flexibility as neurons models have to be specified a priori.

As it was discussed, lowering the resolution of the weight network will result in a higher energy efficiency and lower flexibility of the VMM engine. High-resolution weight network VMM platforms are more vulnerable to the impact of device-level non-idealities like device-to-device and cycle-to-cycle variations. To mitigate the impact of device nonidealities, in addition to device engineering to improve the intrinsic characteristic of the RS memory cells, several circuit- and system-level solutions have been reported. Some of the main ones are using differential weights, WRITE and READ-verify method (closed-loop tuning), applying hybrid training (in situ and ex situ) to reduce the impact of the faulty cells, and nonidealities on the network’s performance. Also, other solutions exist, 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 LSB, and distinguishing between critical and non-critical weights to reduce the impact of faulty cells. There are several other nonideality issues like conductance state-drift, hard faults (stuck-ON and stuck-OFF), etc, which impact the performance of the network. Existing solutions to the conductance state-drift problem include periodical weight reprogramming and feedback designs which have limitations of 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 NNs are ML models implemented with simple digital activation functions (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. The 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 NNs that operate with low-resolution synapses and digital action potentials. Including the time-encoding strategy used in biological networks into binarized NNs could lead to an interesting physical implementation of bioinspired computing. This strategy could potentially reconcile energy efficiency and flexibility of the biological computing system that are still the most inspiring objectives for future hardware development.
Acknowledgements

This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) HIDATA project 506289-2017 and ERC-CoC IONOS N° de convention de subvention 773228. The 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. The authors also 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.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

complementary metal–oxide–semiconductor, inference, in-memory computing, memristors, redox-based random access memories, resistive switching memories, vector-matrix multiplications

Received: May 29, 2020
Revised: July 8, 2020
Published online: August 23, 2020

[1] G. E. Moore, Proc. IEEE 1998, 86, 52.
[2] M. Horowitz, in 2014 IEEE Int. Solid-State Circuits Conf. Digest of Technical Papers (ISSCC), IEEE, Piscataway, NJ 2014, pp. 10–14.
[3] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, B. Khailany, M. Garland, D. Glasco, in 2011 IEEE Symposium on VLSI Technology (VLSI-TS), IEEE, Piscataway, NJ 2011, pp. 365–376.
[4] J. Von Neumann, IEEE Ann. Hist. Comput. 1993, 15, 27.
[5] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, D. Glasco, IEEE Micro 2011, 31, 7.
[6] Y. LeCun, Y. Bengio, G. Hinton, Nature 2015, 521, 436.
[7] K. Simonyan, A. Zisserman, in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Piscataway, NJ 2015, pp. 1–8.
[8] S. Jain, A. Ranjan, K. Roy, A. Raghunathan, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 2017, 26, 470.
[9] D. Ielmini, H.-S. P. Wong, Nat. Electron. 2018, 1, 333.
[10] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, E. Eleftheriou, Nat. Electron. 2018, 1, 246.
[11] D. Strukov, G. Indiveri, J. Grollier, S. Fusi, Nat. Commun. 2019, 10, 4838.
[12] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, E. Eleftheriou, Nat. Nanotechnol. 2020, 15, 529.
[13] A. Haj-Ali, R. Ben-Hur, W. Wald, R. Ronen, S. Kravtinsky, IEEE Trans. Circuits Syst. I 2018, 65, 4258.
[14] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, J. P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, Q. Xia, Nat. Electron. 2018, 1, 52.
[15] S. Liu, Y. Wang, M. Fardad, P. K. Varshney, IEEE Circuits Syst. Mag. 2018, 18, 29.
Amirali Amirsoleimani received a Ph.D. in electrical and computer engineering from the University of Windsor, Ontario, Canada, in 2017. He joined as a postdoctoral fellow to intelligent sensory microsystems laboratory (ISML) at Electrical and Computer Engineering Department of University of Toronto, Ontario, Canada, in 2018. He is still working in ISML lab and his research is focused on in-memory computing hardware design using resistive switching memory for edge computing and brain-machine interface applications.
Fabien Alibart received a Ph.D. in material science from University of Picardie Jules Verne, France, in 2008. He joined IEMN-CNRS in 2012 as a permanent researcher where he worked on the concepts of neuromorphic/bioinspired computing with emerging memory technologies. He is now with LN2-3IT CNRS as a researcher participating in the joint laboratory program between France and Quebec (UMI-CNRS) where he is developing neuromorphic hardware for a variety of applications, from edge computing to brain–machine interfaces.