Device and Circuit Architectures for In-Memory Computing

Daniele Ielmini and Giacomo Pedretti*

With the rise in artificial intelligence (AI), computing systems are facing new challenges related to the large amount of data and the increasing burden of communication between the memory and the processing unit. In-memory computing (IMC) appears as a promising approach to suppress the memory bottleneck and enable higher parallelism of data processing, thanks to the memory array architecture. As a result, IMC shows a better throughput and lower energy consumption with respect to the conventional digital approach, not only for typical AI tasks, but also for general-purpose problems such as constraint satisfaction problems (CSPs) and linear algebra. Herein, an overview of IMC is provided in terms of memory devices and circuit architectures. First, the memory device technologies adopted for IMC are summarized, focusing on both charge-based memories and emerging devices relying on electrically induced material modification at the chemical or physical level. Then, the computational memory programming and the corresponding device nonidealities are described with reference to offline and online training of IMC circuits. Finally, array architectures for computing are reviewed, including typical architectures for neural network accelerators, content addressable memory (CAM), and novel circuit topologies for general-purpose computing with low complexity.

1. Introduction

Data processing in digital computers is generally carried out by a sequence of Boolean logic operations executed in silicon by the complementary metal-oxide-semiconductor (CMOS) technology. The CMOS transistor has been regularly scaling for the last 40 years via Moore's law, where the reduction of the transistor size results in less area consumption, hence lower fabrication cost. Transistor size scaling was accompanied by a reduced power consumption and an increase in the operation frequency, thus leading to an improvement in circuit performance generation after generation.[1] The increase in CMOS logic performance has been challenged by the increase in data processing need and is even more stressed by the exponential growth of data circulating in the internet and provided by always-on and ubiquitous sensors. Unfortunately, reducing the device area also causes an increase in power density which has caused a slowing down in the CMOS scaling trend in the last decade.[2] Conducting AI learning tasks is also heavily demanding in terms of energy consumption, which causes a world-scale concern in view of ubiquitous AI tasks such as image tagging, traffic monitoring, and vocal assistants.[3,4]

Compared with digital computers, the human brain only uses the extremely low power (about 20 W) and low frequency (typically in the few Hz range) of information processing.[5] The human brain thus appears as a living biological example to help introduce novel energy-efficient computing paradigms to tackle data-intensive and AI tasks. One of the main assets of the human brain which enables low energy consumption is its peculiar architecture, where memory and computation are colocated.[6] This is against the conventional computer architecture, where computing takes place in a central processing unit (CPU) according to programs and data which are fetched from a working memory according to the von Neumann architecture.[7] The working memory, i.e., most typically a dynamic random-access memory (DRAM), is generally located on a physically separate chip, thus resulting in long latency and energy consumption for data intensive tasks. Similar to the human brain, in-memory computing (IMC) instead conducts data processing in situ within a suitable memory circuit.[8] IMC suppresses the latency for data/program fetch and output results upload in the memory, thus solving the memory (or von Neumann) bottleneck of conventional computers. Another key advantage of IMC is the high computing parallelism, thanks to the specific architecture of the memory array, where computation can take place along several current paths at the same time. IMC also benefits from the high density of the memory arrays with computational devices, which generally feature excellent scalability and the capability of 3D integration. Finally, analogue computing is supported by the physical laws of memory circuits, such as the Ohm's law for product and the Kirchhoff's law of current summation,[8–11] as well as other memory-specific physical behavior such as nonlinear threshold-type switching, pulse accumulation, and time measurement.[12–15] Thanks to the combination of in situ, high-density, parallel, physical, and analogue data processing, IMC appears as one of the most promising novel approaches for computing in the frame of AI and big data.

Prof. D. Ielmini, Dr. G. Pedretti
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano and Italian Universities Nanoelectronics Team (IU.NET)
Piazza L. da Vinci 32, 20133 Milano, Italy
E-mail: daniele.ielmini@polimi.it

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202000040.

DOI: 10.1002/aisy.202000040

In addition to analogue computing, digital-type IMC supported by physical properties of the memory devices has also been shown. Stateful logic can be realized in a memory circuit by comparing the voltages among two or more devices and conditionally inducing switching in one of these devices or in an additional one, by the threshold-dependent set/reset operations.[12,14–19] This approach can improve the density of logic gates and suppress the latency associated with data transfer for digital computation. On the other hand, digital IMC suffers from an increased energy per operation due to the need to change the state of a device during computation. The state switching also increases the time for logic operation and critically limits the lifetime of the circuit due to endurance constraints. For these reasons, a device technology breakthrough might be needed to support the development of largely scaled, low-energy, high-performance logic IMC processors.

This work presents an overview of IMC in terms of device technologies and circuit architectures. Within the extremely large scenario of IMC concepts, we focus our attention on analogue-type computing based on matrix-vector multiplication (MVM) in the memory array. In Section 2, we provide an overview of devices for IMC, covering both two-terminal and three-terminal devices that have emerged recently. In Section 3, we describe the main memory structures which are used in IMC circuit. In Section 4, we focus on the programming operation, where a certain set of conductance values are stored in the memory circuit to serve a certain IMC operation. In this respect, we describe the main methodologies to program a set of conductance values in the computational memory to serve for a certain IMC function. In this respect, we highlight the main programming methodologies as well as the most typical nonidealities which affect the accuracy of the IMC operation during either the offline or online training of the memory array. Section 5 address the nonidealities of the memory circuit. Finally, Section 6 presents the main architectures that have been proposed for IMC, including crosspoint arrays and other computational memory arrays, which are relevant for various types of neural networks and general-purpose algebraic computing tasks.

2. Memory Devices for IMC

Recently, several memory technologies based on the material modification at the nanoscale have emerged as high-density, low-power, low-cost, and high-speed devices for storage and computing.[7,8,15,20,21] In general, the material modification, such as a local change in the chemical composition or phase structure, causes a major change in the device resistivity which can be easily sensed by the peripheral circuit via electrode wires. In particular, these two-terminal devices offer the advantage of scalability to only few nm[22–24] and integration in 3D,[25,26] thus supporting the ultrahigh density of memory needed for computing applications.

Daniele Ielmini is a full professor at the Dipartimento di Elettronica, Informazione, e Bioingegneria of Politecnico di Milano, Politecnico di Milano. He received his Ph.D. degree from Politecnico di Milano in 2000. He conducts research on emerging nanoelectronics devices, such as PCM and RRAM, and on novel computing with memory devices.

Giacomo Pedretti received his B.S., M.S., and Ph.D. (cum laude) degrees in electronics engineering from Politecnico di Milano, Milan, Italy, in 2013, 2016, and 2020, respectively, where he is currently a postdoc research associate. His research interests include the design of memristive circuits for optimization and analog computing.

Figure 1 shows a summary of two-terminal devices which are currently considered for storage and computing. Device technologies include the resistive-switching random access memory (RRAM), the phase-change memory (PCM), the magnetic random-access memory (MRAM), and the ferroelectric random-access memory (FERAM).

2.1. RRAM Devices

Figure 1a shows the RRAM device, consisting of a stack of metallic top electrode (TE), an insulating metal-oxide layer, and a metallic bottom electrode (BE).[27–29] The resulting metal-insulator-metal (MIM) structure shows a relatively large resistance, thanks to the insulating nature of the oxide layer. This is sometimes replaced with an alternative high-resistance material,

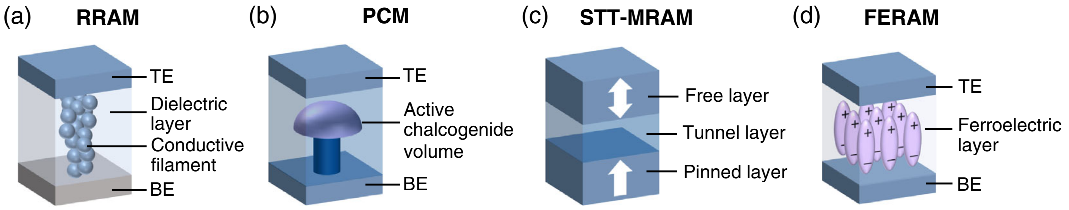

Figure 1. Illustration of the two-terminal memory devices for storage and computing. a) RRAM, where the device resistance is controlled by field-modulated filamentary paths in the dielectric layer. b) PCM, where the device resistance is controlled by the amorphous/crystalline phase in the chalcogenide active layer. c) STT-MRAM, where the device resistance is controlled by the parallel/antiparallel polarization of the ferromagnetic layers in the MTJ. d) FERAM, where the electrostatic polarization is controlled by the orientation of FE domains in the ferromagnetic active layer.

such as a nitride layer,[30] a chalcogenide material,[31–33] or 2D transition metal dichalcogenides (TMDs).[34] The MIM device is first electrically formed by a soft breakdown operation, causing a local modification of the material composition or an increase in the defect concentration, such as oxygen vacancies in the metal oxide. The forming operation generally causes the buildup of a conductive filament, where the conductance is higher than that in the original insulating layer, thus resulting in a low resistance state (LRS) of the device. The conductivity of the conductive filament can be electrically reduced by the reset operation, leading to a high resistance state (HRS) of the device, or increased by the set transition, to recover the LRS. In a bipolar RRAM device, the set and reset transitions are induced by voltage pulses of opposite polarities, whereas the polarity of set/reset operations is the same in unipolar RRAM devices.[35] Uniform-switching RRAM devices also exist where the oxide layer modification extends throughout the whole area instead of a localized filament region.[36,37]

2.2. PCM Devices

Figure 1b shows the PCM device, where the microstructure of a phase-change material, generally a chalcogenide material such as $Ge_2Sb_2Te_5$ (GST), can be reversibly switched between a crystalline phase and an amorphous phase.[38–41] The amorphous phase shows a disorder-induced high resistivity, in contrast with the low resistivity of the crystalline phase; thus, the PCM state can be identified by a simple voltage/current sensing. Compared with the filamentary switching process of the RRAM, the PCM relies on the bulk properties of the active material, which generally leads to a larger resistance window and the ability to operate the device with a multilevel cell (MLC) scheme.[42,43] On the other hand, a large Joule heating is generally needed to accelerate the phase transitions, such as melting and crystallization, which result in relatively large currents for programming/erasing the device. Reducing the programming current requires the scaling of the active region of the PCM.[44–46] A significant problem for the PCM is the resistance drift, where the device resistance increases with time after programming due to the structural relaxation of the amorphous phase.[47] Device technologies with improved stability against drift have been developed[48] and demonstrated in IMC.[49]

2.3. MRAM Devices

Figure 1c shows the MRAM device, where the magnetic polarization within a layer of ferromagnetic material such as CoFeB is changed by electrical manipulation. The residual polarization in the ferromagnetic material can be sensed via the magnetic tunnel junction (MTJ), namely, a stack made of a thin insulating layer, usually a highly crystalline metal oxide such as MgO, sandwiched between a reference ferromagnetic layer with fixed polarization and a free ferromagnetic layer with variable polarization. When the two layers have parallel magnetization directions, the resistance of the MTJ is relatively low, whereas the MTJ resistance is relatively high for antiparallel magnetization.[50] The magnetization direction in the free layer can be written by field-induced switching, where a current pulse is applied across suitable write lines to create a local magnetic field,[51] or spin-transfer torque (STT), where the current pulse is applied directly across the MTJ.[52,53] STT-MRAM devices have the advantage of fast switching in the few ns range, which makes them a strong candidate for last-level cache (LLC) static RAM.[54] On the other hand, MRAM generally displays a limited resistance window around a factor 2, which makes it difficult to implement some IMC algorithms.[55]

2.4. FERAM Devices

Figure 1d shows the FERAM device concept, which is based on ferroelectric (FE) materials where the electrostatic polarization can be reversibly switched by the application of an external electric field. Historically, most typical FE materials include perovskite oxides such as $PbZr_{1\text{-}x}Ti_xO_3$ (PZT) and $SrBi_2Ta_2O_9$ (SBT).[56] These materials, however, have relatively a low bandgap, high leakage, and low compatibility with the CMOS process line. Most recently, FE phases of doped HfO_2 have been discovered,[57] which have revived the interest on FE phenomena and materials for both storage and IMC applications. Similar to the MTJ, an FE tunnel junction (FTJ) is able to convert a residual FE polarization into a resistance signal, by placing the FE switching layer in series with a dielectric layer.[58,59] The FTJ structure can be easily programmed by application of voltage pulses. Despite the nonfilamentary switching within the FE layer, FERAM uniformity can be affected by local variation in the coercive fields among various crystalline grains and domains within FE material.[60]

2.5. Three-Terminal Devices

Although the two-terminal structure is strongly promising for crosspoint architectures with high densities, three-terminal devices might tradeoff density with other properties such as a better control of the conductance state or an easier cell selection within the array. **Figure 2** shows a summary of three-terminal devices that have been considered for IMC. The Flash device (Figure 2a) is at the basis of most nonvolatile memory devices used for high-density storage in solid-state drives (SSDs). The Flash memory essentially consists of a metal-oxide-semiconductor (MOS) transistor with a floating gate (FG) between the contacted gate and the substrate. The charge stored in the FG can be electrically manipulated by high-field tunneling of electrons to/from the substrate.[61] Once stored in the FG, the charge affects the transistor threshold voltage, namely, a larger amount of electrons in the FG results in a higher value of the threshold voltage. Alternatively, a different amount of charge also corresponds to a different channel conductance, that can be used as a variable resistance for IMC. This concept was used for hardware accelerators of neural networks with arrays of Flash memories.[62] Also, unsupervised learning by spike-timing-dependent plasticity (STDP) was demonstrated with Flash memories.[63,64]

Figure 2b shows the typical structure of a DRAM, which represents the standard device for working memory in digital computers. Different from the Flash memory, in a DRAM, the charge is stored at a capacitor at the gate of the conduction transistor. Another pass transistor is generally kept in the off state, unless during the programming operation, when the pass transistor is switched on. The charge across the capacitor can be tuned to

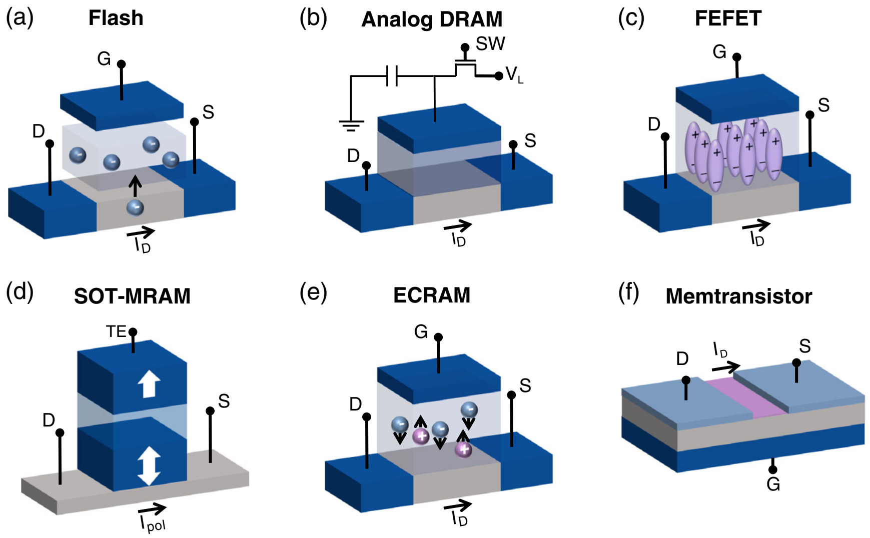

Figure 2. Illustration of three-terminal memory devices for storage and computing. a) Flash, where the threshold voltage of the transistor is controlled by the charge stored within the FG. b) Analog DRAM, where threshold voltage of the transistor is controlled by the charge stored across an independent capacitor. c) FEFET, where the threshold voltage is controlled by the orientation of the FE dipoles within the gate insulator. d) SOT-MRAM, where the MTJ resistance can be electrically manipulated by the in-plane current I_{pol} along a HM line. e) ECRAM, where the channel conductance is manipulated by the field-induced ionic migration. f) Memtransistor, where the conductance is controlled by the migration of defects across a 2D semiconductor channel.

control the threshold voltage and conductance of the conduction transistor for analogue IMC.[65] To increase the retention time of the capacitor charge, which is around 1 ms in DRAM, the pass transistor can be fabricated with low-mobility semiconductors such as InZnGaO.[65] The lower subthreshold channel conductance helps enhancing the retention time so that the analogue DRAM can be used for practical IMC applications.

The ferroelectric field-effect transistor (FEFET), shown in Figure 2c, is a transistor concept where the threshold voltage is varied by the remnant polarization in the FE gate-insulating layer.[66,67] FEFETs can be arranged with NAND architecture in either 2D[68] or 3D,[69] which may allow to reach similar density as Flash memories. The interest in FEFET has significantly increased after the discovery of FE phases in HfO_2,[57] thanks to the better CMOS compatibility of HfO2 with respect to FE ternary/quaternary oxides.

Figure 2d shows the spin-orbit torque (SOT) MRAM, consisting of an MTJ deposited on top of a heavy metal (HM) line such as Ta[70] or Pt.[71] In the STO-MRAM, the parallel/antiparallel states of MTJ can be manipulated by applying an in-plane current pulse along the HM line via SOT induced by spin Hall or Rashba effect. Sub-ns switching speed has been demonstrated with current densities in the range of few hundreds of $MAcm^{-2}$.[71] The main advantage of SOT-MRAM with respect to the STT structure is that the programming operation does not involve any current across the MTJ, which was the major source of degradation and endurance failure in STT-MRAM. This advantage comes at the price of a three-terminal structure, hence a larger device area. Similar to STT-MRAM, the SOT-MRAM also typically shows binary switching between the parallel and the antiparallel state, which is not suitable for analogue-type IMC.

Figure 2e shows the ionic transistor, also known as the electrochemical random access memory (ECRAM). In a Li-based ionic transistor, the gate dielectric consists of an ionic conductor for Li^+ such as lithium phosphorous oxynitride (LiPON).[72] The channel transistor instead consists of a material such as $LiCoO_2$ where Li^+ intercalation and deintercalation can induce a change in channel conductivity. For instance, the application of a positive gate voltage leads to Li^+ migration and channel lithiation, which leads to a reduction in conductivity.[72] Ionic transistors have also been developed based on organic materials where H^+ was the migrating ion.[73] The Li-based ionic transistor has shown a strong linearity where an applied gate pulse causes a fixed increase or decrease in conductivity.[74] A potential problem of the Li^+-based synaptic transistor is the leaky gate, due to the relatively high conductance of the solid-state electrolyte. To prevent the corresponding leakage, a selector device has to be connected to the gate of the ionic transistor, which significantly increases the array complexity.[75] Another potential issue is the lack of compatibility with the CMOS process line, for which Li^+ is considered a concern. To solve both these issues, recently, a metal-oxide-based ionic transistor was proposed.[76] In this device, the migration of oxygen vacancies across a trilayer metal oxide causes a change in the conductance of the WO_3 channel. Thanks to the insulating property of the metal-oxide stack, no selector is needed in series with the gate.

Figure 2f shows the memtransistor, a contraction of memristive transistor, consisting of a MOS transistor with a 2D semiconductor channel, such as MoS_2.[77] In this structure, the application of a large source-drain voltage leads to a permanent change in conductivity as a result of the migration of grain boundaries[78] or Li^+ impurities in the MoS_2 channel.[79] The gate can be used to control the channel conductance, e.g., to activate and deactivate the defect migration induced by the source–drain voltage. The use of a 2D semiconductor makes the memtransistor highly scalable and suitable for 3D integration in the back end.

3. Memory Structures

Figure 3 shows the possible memory array structures for two-terminal devices. In the one-resistance (1R) structure (Figure 3a), the memory device is tied to a row wire by the TE and a column wire by the BE or vice versa. This is the conventional crosspoint array,[15] which allows the maximum density of packing memory devices on the chip. The minimum theoretical area for the 1R device is $4F^2$, where F is the lithographic feature size, which dictates the width of the row/column and their spacing. This density can be further increased in case of the 3D stacking of more crosspoints.[80] For instance, the effective device area becomes $2F^2$ for a two-layer crosspoint and only F^2 for a four-layer crosspoint. Both horizontal stacking of crosspoint arrays and vertical arrays can be realized, the latter achieving a higher density, thanks to the increased stackability due to the easier patterning process of vertical wires.[81] Thanks to the close packing of the crosspoint structures, the memory density of 4.5 Tb per square inch has been demonstrated in one layer.[24]

Assuming that conductance values G_{ij} are stored in the memory devices at row i and column j, the application of column voltages V_{j} will induce a current $G_{ij}V_{\mathrm{j}}$ in each device, according to the Ohm's law. Based on the Kirchhoff's law, the row current reads

$$I_i = \sum_j G_{ij} V_j \tag{1}$$

which can be expressed in vectorial form as $\mathbf{I} = G\mathbf{V}$, where I is the current vector, G the matrix of conductance values stored in the array, and $\mathbf{V}$ is the voltage vector.[8–11] The passive crosspoint array is thus capable of executing a parallel MVM in the analogue domain, which would require instead a huge number of multiply-accumulate (MAC) operations in a conventional digital computer.

During MVM, voltages are applied simultaneously to all columns whereas currents are collected at the grounded row terminals. Ideally, assuming negligible voltage drop as a result of parasitic wire resistances, the MVM operation should not suffer from any cell–cell disturb or sneak-path effect.[82] On the other hand, when individual devices are programmed, such as for executing forming, set, and reset operations in the array, disturbs might become a significant problem. For instance, application of a positive voltage at a certain column of a crosspoint array of RRAM devices might potentially induce set operation on all cells in the row, unless specific biasing schemes are adopted.

A typical approach to overcome potential disturbs during array programming is the $V/2$ scheme shown in Figure 3b.[83–85] In this biasing scheme, voltages $V/2$ and $-V/2$ are applied to the selected column and row, respectively, whereas all other lines are grounded. As a result, the bias voltage across the selected cell is V, whereas all other unselected cells are biased at 0 V, and half-selected cells, sharing the same row or column of the selected cell, are biased at $V/2$ or $-V/2$. As a result, the voltage drops across nonselected and half selected is significantly lower than

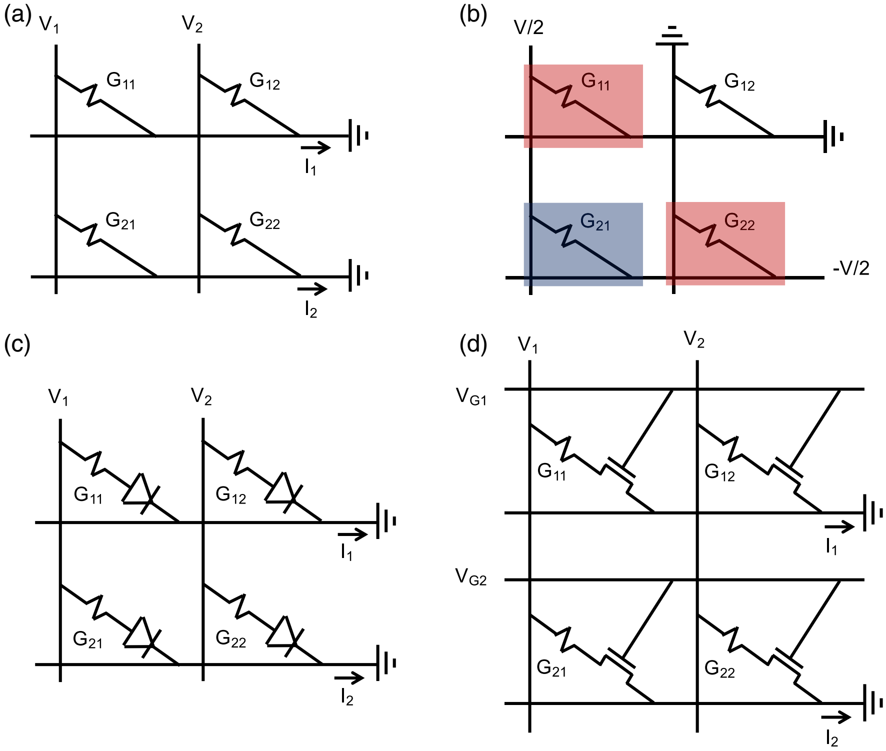

Figure 3. Illustration of memory array structures for two-terminal devices. a) Passive crosspoint array, consisting of 1R elements with conductance G_{ij}, each connected between a row and a column. b) $V/2$ biasing scheme for 1R arrays, where a voltage V is applied across the selected cell (blue), whereas half-selected cells (red) sharing the row/column of the selected cells are biased at voltage $V/2$. c) 1S1R array, where each memory element is connected to an individual selector to prevent sneak paths. d) 1T1R array, where the select transistors allow to select a cell at the crossing between the selected wordline and bitline.

the one across the selected cell, thus preventing any disturb within the array. To read an individual cell a voltage V_R is applied to the cell row, whereas all other rows and all columns are grounded. The current at the selected column will reveal the resistance R of the selected cell according to $I = V_R/R$.[85] Note that reading individual cells is essential to make sure that a conductance value G_{ij} is stored correctly in the crosspoint array.

Although the set, reset, and read operations appear feasible with the $V/2$ biasing scheme, the 1 R crosspoint architecture becomes unpractical because of the large standby current flowing during set and reset. Also, as the selected device is being set at V, the voltage $V/2$ and the corresponding current flowing across half-selected devices in the LRS might be sufficiently high to disturb the device, thus modifying the previously stored conductance.

3.1. 1S1R Structure

To solve the issues in the 1R crosspoint array, the one-selector/one-resistor (1S1R) structure in Figure 3c can be adopted.[86–88] In this structure, the memory device is connected to a selector device with a strongly nonlinear I–V characteristic, where the current is virtually zero below a threshold voltage V_t. As a result, as a voltage $V > V_t$ is applied to the selected cell to induce set/reset processes, the half-selected voltage $V/2 < V_t$ will not induce any disturb. Both silicon-based and nonsilicon-based selectors have been proposed, the latter category being favored as it enables the back-end-of-line (BEOL) process and 3D stacking. Various nonsilicon selector concepts have been proposed, including oxide-based p–n diodes,[89,90] oxide-based tunneling layers,[91,92] Mott oxides with insulator-metal transition,[93] mixed ionic–electronic conductors (MIEC),[94] and ovonic threshold switching (OTS) materials.[95–97] OTS selectors are characterized by the low subthreshold leakage, large V_T, and negative differential resistance (NDR), which allows an excellent nonlinearity factor of several orders of magnitude between the off-state and on-state currents. In addition, OTS shows good endurance of above 10^{11}[98] and the ability for stacking at least two layers.[95,97] The 1S1R concept is very promising for creating a new memory market named storage-class memory (SCM), combining nonvolatile storage, a density higher than DRAM, and a performance better than Flash memories. Because of these properties, the 1S1R structure seems an ideal vehicle for IMC applications, although the nonlinear behavior of threshold-switching selectors and the corresponding large current in the on state have to be carefully considered in the architecture design.

3.2. 1T1R Structure

Figure 3d shows the one-transistor/one-resistor (1T1R) structure, where the memory device is connected to an MOS transistor for selection. With respect to the 1R and the 1S1R structures, the 1T1R structure is more complicated in that a third terminal and a corresponding wire must be dedicated to the transistor gate. The presence of the gate terminal makes the selection and unselection of the array device extremely straightforward. The gate line is perpendicular to the TE line; therefore, only the device at the intersection between the selected gate line and the selected TE line is addressed during set, reset, and read. In addition, the transistor allows for a proper current limitation during forming and set transition of RRAM devices to control the resistance state of the LRS.[99,100] During reset and read, instead, the gate terminal is biased to a relatively high voltage to reduce the parasitic resistance of the MOSFET, which might degrade the precision and dynamic range of the conductance G for analogue MVM. The larger flexibility, however, comes with the expense of a larger device area and higher complexity of the array. Despite these drawbacks, the 1T1R structure is by far the preferred structure for IMC applications.

The circuit structures of Figure 3 are limited to two-terminal devices, although the 1T1R structure can be adapted for three-terminal devices, such as three-terminal Flash memory array. This is the so-called NOR structure, where applying a pulse at a given gate (word) line and a given drain (bit) line results in the programming of the device, without affecting all other devices in the array. In general, however, dedicated array structures might be needed for correct programming, reading, and computing with three-terminal devices.

4. Computational Memory Programming

One of the strongest advantages of IMC is the ability to parallelize analogue MVM within a memory array, according to Figure 3a. The most straightforward application of MVM is the realization of a hardware neural network, where the synaptic weights can be stored as the memory conductance within the array.[8–11,100] Each layer of the network can be thus mapped into a memory array, where each memory element stores a synaptic weight. On the other hand, nonlinear activation functions are generally achieved by an external analog or digital circuit. Similarly, memory-based MVM in the crosspoint array can accelerate other types of computations, such as linear algebra and image processing.[10] For all these IMC applications, which we refer to as "computational memory," the device requirements are different from those of a simple memory, in at least three aspects. First, a high precision in the stored conductance values G_{ij} of the computational memory is essential, to compete with floating-point precision of digital MAC. While such a strong precision of conductance is not strictly necessary for memory or storage applications, which are generally limited to 1- or 2-bit precision, the analogue-type accuracy of conductance is instead a key requirement for IMC. The second requirement is that of a relatively high resistance, to limit the overall summation of all the individual computational memory currents according to Kirchhoff's law. In fact, a large current would result in a large size of the transistor for column selection. To reduce the current, each computational memory device should have a relatively high resistance, which would also help reducing the parasitic voltage drop across the array rows/columns. On the other hand, the read current for memory applications should be a large as possible, to enable fast random readout and easy design of the sense amplifiers (SAs). The third difference which distinguishes computational and conventional memories is the required performance in terms of the programming time. The programming time for a computational memory element is generally relaxed with respect to the case of the conventional memory, as programming must be operated only at the beginning of the IMC operations and state reconfiguration is generally rare. This is the case of “offline training,” where conductance G_{ij} that has to solve a certain task is stored at time zero in the memory array and later reconfigured only if/when needed. The values G might consist of either input data obtained from sensors, e.g., the genes from a DNA sequencer or synaptic parameters obtained from the backpropagation algorithm to train a fully connected neural network.

Opposite to offline training, the “online training approach” consists of iteratively adjusting the memory conductance directly on the hardware memory array, e.g., by adopting standard gradient descent techniques such as the backpropagation algorithm. This approach allows to take advantage of the IMC energy benefit in both the training and the inference tasks.

4.1. Offline Training

To address offline training procedures and the corresponding sources of nonideality, we consider a RRAM device with 1T1R structure.[101] **Figure 4**a shows the *I–V* characteristics of the RRAM device for increasing gate voltage of the select transistor. The RRAM device consists of an active HfO_2 layer sandwiched between a Ti TE and a C BE, the latter connected to the drain of the transistor according to the structure in the inset of Figure 4. Set transition takes place as the applied voltage across the 1T1R structure reaches a characteristic voltage V_{set} of about 2.2 V. During set transition, the gate voltage controls the saturated transistor current, which in turn controls the final conductance of the LRS.[99] Then, the application of a negative voltage causes the reset transition to the HRS.

From the results in Figure 4a, the gate voltage appears the most suitable parameter to control the conductance G of the RRAM for IMC applications. This is shown in Figure 4b, showing the measured G after the application of a set pulse with increasing gate voltage V_G.[101] The individual traces for 100 experiments from the same device are shown and compared with the average conductance. The average conductance increases almost linearly with $V_G - V_T$, where $V_T = 0.7$V is the threshold voltage of the transistor. However, the individual traces display noisy characteristics due to the stochastic ionic migration during the physical set process.[102,103] Figure 4c shows the distributions of G for increasing V_G, indicating a normal shape with a standard deviation $\sigma_G = 3.8\,\mu S$, independent of the programming level. These results suggest that accurate programs/verify algorithms are needed to correctly tune the conductance for IMC.

In addition to the cycle-to-cycle variability displayed by individual devices, there is also a device-to-device variability arising from differences in the composition, structure, and geometry of the cells within the array. Figure 4d shows the distributions of read current at $V_{read} = 0.5$ V for RRAM cells with the HfO_2 switching layer, which were programmed with four different levels (L2–L5) of compliance current.[100] The lowest current level L1 corresponds instead to the HRS. All distributions show a significant

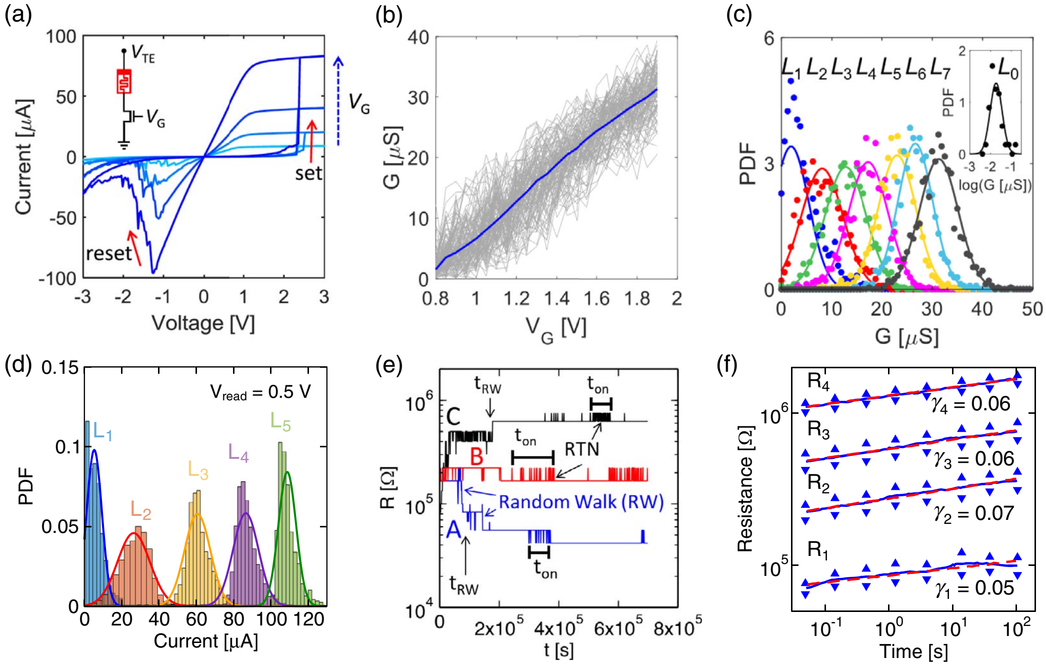

Figure 4. Offline training of a computational RRAM. a) *I–V* characteristics of a bipolar RRAM with 1T1R structure (inset) for increasing gate voltage of the select transistor. V_G controls the compliance current I_C during set transition, hence the device conductance G. b) Measured RRAM conductance as a function of the gate voltage V_G, indicating an almost linear increase in average behavior. Note the relatively large cycle-to-cycle variations of G. c) Distributions of conductance for seven levels of LRS and one level of HRS (inset). The standard deviation is $\sigma_G = 3.8\,\mu S$, independent of the programmed level. a–c) Reproduced with permission.[101] Copyright 2020, IEEE. d) Cell-to-cell distributions of measured current at $V_{read} = 0.5$ V in an 1T1R array for five programmed levels. Reproduced with permission.[100] Copyright 2019, AIP Publishing. e) Time-dependent fluctuations of resistance R for a RRAM device in HRS, indicating both RW and RTN phenomena. Reproduced with permission.[110] Copyright 2015, IEEE. f) Resistance drift of a PCM device programmed at four levels. Adapted with permission.[48] Copyright 2013, IEEE.

variation in current, which includes both cycle-to-cycle and device-to-device contributions.

Programming variability effects can be alleviated or even suppressed by accurate program-verify techniques. For instance, in the scheme of Figure 4b, one might gradually increase the gate voltage to reach a certain target G. If G is exceeded by an error ΔG which exceeds the tolerable window, corresponding, e.g., to an accuracy of 8 bits, then the device can be reinitialized to the HRS and a new V_G ramp is attempted. Instead of restarting from the HRS, one might apply suitable negative voltage pulses to gradually decrease G, until the error become smaller than the tolerance.[104,105] This approach takes advantage of the RRAM being able to gradually increase and decrease G in RRAM devices by application of positive and negative voltage pulses, respectively. Despite the energy and time needed to conduct, such as an accurate program-verify technique may be considerable, the overhead might still be tolerable, as long as the device conductance is not frequently updated. For instance, some memory arrays might be programmed only once for neural network accelerators, so that the programming time/energy might be amortized over the whole chip lifetime.

An extreme case of device-to-device variation is the possibility that the memory cell is stuck to a nonideal state, such as a LRS, a HRS, or an intermediate state, for instance, as a result of the cycling endurance failure.[106–108] Another possibility is that the RRAM device cannot be formed, thus resulting in an extremely low value of G, even lower than the HRS value. In all these cases, it is clear that, in most cases, the matrix G_{ij} cannot be stored correctly in the memory array. These problems can be solved with suitable redundancy schemes, where the individual cell, or most typically its entire row/column, are disabled and replaced by a spare one. Error-tolerant online training schemes have also been proposed to correctly compensate these stuck memory elements.[109]

Even if the programming operation appears successful at time zero, the conductance might still change after the programming step as a result of subsequent relaxation or fluctuation of the microscopic structure of the device. Figure 4e shows a typical fluctuation of resistance, following a reset pulse on an RRAM device.[110] Three devices with the same initial resistance were chosen initially and measured at increasing time. The devices show abrupt steps of resistance, called random walk (RW) and random telegraph noise (RTN). As a result, the cell resistance can increase, decrease, or stay unchanged.

Another typical phenomenon of unstable resistance is the drift process of PCMs. Figure 4f shows the measured resistance of PCM as a function of time after the reset process for four different levels of an MLC. Various resistance levels in the PCM can be obtained, e.g., by amorphizing an increasing volume of the PCM.[111] The resistance increases with the amount of amorphous volume in the PCM, as the amorphous phase has a higher resistivity than the crystalline one. The PCM resistance increases with time in the figure can be attributed to the structural relaxation of the amorphous phase,[112] consisting of an annihilation of defects, such as Ge—Ge wrong bonds,[113] and the consequent increase in the mobility gap.[114] Both resistance fluctuation and drift clearly represent significant problems for analogue MVM, where the conductance G of all elements in the array should remain stable.

4.2. Online Training

Figure 5a shows a typical three-layer multiple-layer perceptron (MLP), where input signals propagate from left to right. In the forward propagation, a neuron n_j of a generic layer generates a signal x_j that is sent out to all output neurons m_i in the next layer after multiplication with the synaptic weights w_{ij} connecting neuron n_j with neuron m_i. The signal received by any neuron m_i is given by the accumulation of all weighted signals from the previous layer, which thus reads

$$y_i = \sum_j w_{ij} x_j \tag{2}$$

This formula perfectly matches Equation (1), namely the analogue MVM executed by the memory array of Figure 3a. A neural network can thus be implemented in a memory array, where the MVM at each neuron layer is executed in the analogue domain within a memory array.[8–11,115,116] It has been estimated that, thanks to the suppression of data movement in the IMC architecture, the energy consumption is reduced by more than 10 000 times in an RRAM array with respect to the conventional MAC approach in digital computers.[117] To correctly map a neural network with a memory array, however, the conductance G should be able to implement both positive and negative values of the synaptic weight w_{ij}. To this purpose, two circuits are generally adopted: in the first circuit, the current $I = VG$ is compared with the current $I_{\mathrm{ref}} = VG_{\mathrm{ref}}$, obtained from a reference cell biased at the opposite voltage (Figure 5b). Current comparison can be achieved by simple Kirchhoff's law and the current can be used to feed the activation function of the output neuron, together with all current contributions from other synapses. In this scheme, the effective synaptic weight is given by $G - G_{\mathrm{ref}}$, which can thus be positive or negative depending on the value of G with respect to G_{ref}. In the second circuit, the synaptic weight is mapped by a pair of conductances G^+ and G^-, which are biased at positive and negative voltages, respectively.[115,116] The equivalent conductance is $G^+ - G^-$ which can again have either a positive or negative sign.

The memory array can accelerate not only the forward propagation from input to output layers during the inference mode, but also the so-called backpropagation algorithm for online training.[115,118] In this approach, the synaptic weights are updated after the submission of a whole (or part of the) dataset, and the iterative repetition of the update allows to minimize the error and improve the accuracy of the network. Referring to the network of Figure 5a, the online training process consists of three phases, namely 1) forward propagation, 2) backward propagation, and 3) weight update. In the first operation, an input sample of the dataset is presented at the input and propagated throughout the network, thus leading to results y_j appearing at the output layer. These results are compared to the ideal results o_j, thus yielding a set of errors $\delta_j = y_j - o_j$. At this point, one should backpropagate the error and update the value of each synaptic weight w_{ij}, according to the weight update rule

$$\Delta w_{ij} = \eta x_i \delta_j \tag{3}$$

where, x_{i} is the signal at the synapse during the forward propagation and η is the learning rate.[118,119] In this scheme, the

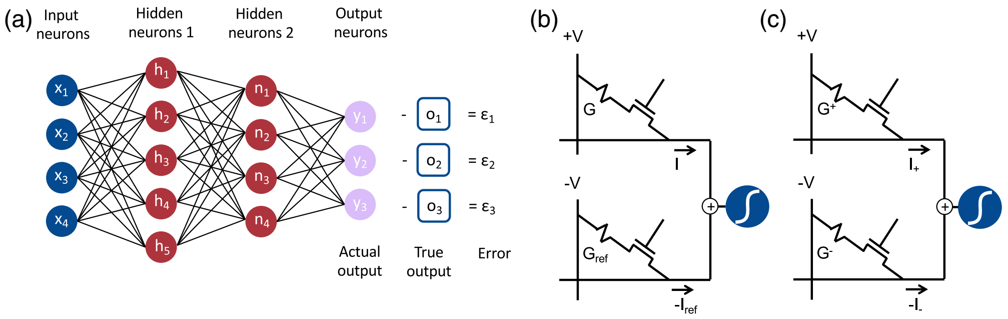

Figure 5. Neural network implementation with memory arrays. a) Schematic illustration of an MLP with three synaptic layers. The output y_1, y_2, etc. is compared with the true output o_1, o_2, etc., to yield the error ε_1, ε_2, etc., which can be backpropagated to perform the training of the network. b). A possible implementation of the synaptic weight by a 1T1R memory, where the current I is compared with a reference current I_{ref} across a common conductance G_{ref}, thus resulting in an equivalent conductance $G - G_{ref}$ to enable mapping of both positive and negative weights. c) Another possible implementation of the synaptic weight with two 1T1R elements, where the synaptic weight is described by equivalent conductance $G^+ - G^-$.

weight must be updated with the least amount of time and energy, for best efficiency of the online training process. Thus, the weight should be updated without any preliminary read or following verify pulse; rather, a single update pulse at fixed voltage and time should be operated.

To test the compatibility of a memory device to online training, the standard approach consists of the application of a train of positive voltage pulses for weight increase, followed by a train of negative voltage pulses for weight decrease. This is shown in **Figure 6** for a typical bipolar switching memory capable of weight update on both positive and negative voltage pulses. Figure 6a shows the ideal behavior of the memory device, where the conductance G increases and decreases linearly for the increasing number of pulses. In this case, the weight update $\Delta G = \Delta w_{ij}$ is constant, irrespective of the initial conductance G, thus allowing for a weight update according to Equation (3) without any preliminary measurement of G. In general, however, memory devices show a nonlinear weight update, such as the one shown in Figure 6b. Here, the initial pulses cause a steep increase in conductance, followed by a saturation at longer pulses. The same occurs for negative pulses. This is the behavior generally observed for bipolar RRAM devices.[116] In this implementation, the synapse can have the structure of Figure 5b where G_{ref} is kept constant, whereas G is increased or decreased to change the overall synaptic weight.

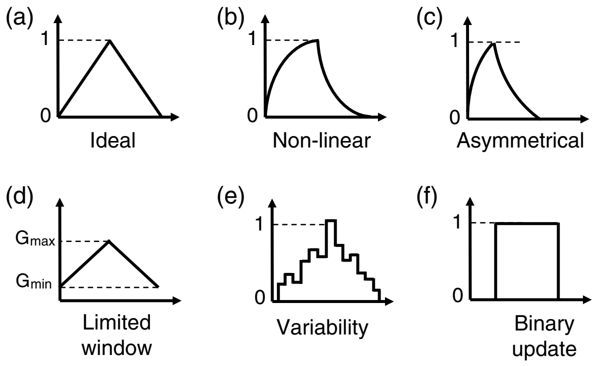

Figure 6. Illustration of the weight update characteristic, namely the device conductance G as a function of the number of pulses of the constant positive voltage and constant negative voltage. a) Ideal characteristic, where G increases linearly with positive voltage pulses and decreases linearly with negative voltage pulses. b) Nonlinear characteristic, where G increases (decreases) steeply first, then saturates at positive (negative) voltage pulses. c) Asymmetric characteristic, where the shape of the response to positive and negative pulses differs significantly. d) Limited window (dynamic range) of G where values close to $G = 0$ cannot be reached. e) Variability of the update characteristic due to cycle-to-cycle variations similar to Figure 4b. f) Binary update, where no gradual update is possible for neither positive nor negative voltage pulses.

In addition to nonlinear update, the weight increase and decrease might also display asymmetric shapes due to different linearity factors for positive and negative applied pulses (Figure 6c). The impact of the asymmetric weight update is that more pulses might be needed to increase the conductance by a contribution ΔG than the number of pulses needed to decrease the conductance by the same amount. There is only one conductance value G_{sym}, in general, where the derivatives of the increase and decrease characteristics are the same.[120] In the zero-shifting technique, the reference conductance G_{ref} is chosen to be equal to G_{sym}, so that the symmetric response is obtained for $G_{ref} = G_{sym}$, corresponding to $G = 0$.[76,120]

An extreme case of asymmetric update is the PCM device, where G can gradually increase via crystallization, whereas the conductance decrease induced by phase amorphization is generally abrupt and nongradual.[118] In this case, the synaptic weight has the structure of Figure 5c, where the crystallization-induced increase in G^+ causes an overall increase in weight, whereas the crystallization-induced increase in G^- causes an overall decrease in weight. A change in G can thus be achieved by unidirectional updates in G^+ and G^-, i.e., an increase of G can be achieved by an increase in G^+ or a decrease in G^-. A significant problem of the unidirectional update scheme is the limited increase in G^+ and G^-, which can never exceed the maximum value corresponding to the fully crystalline state.[118] When one of the two conductances reaches the maximum value, then a reset operation is necessary, to allow for further update operations. For instance, if G^+ reaches the maximum value G_{max}, then both G^+ and G^-should be reduced to keep a constant $G = G^+ - G^-$, while allowing for further increase in G^+.

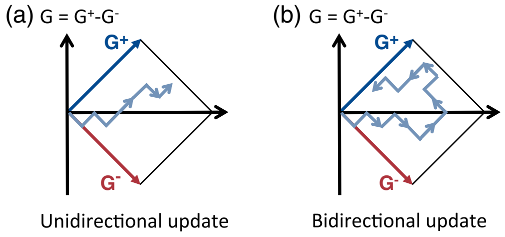

Figure 7. Illustration of the weight update for the differential synaptic memory of Figure 5c. a) Unidirectional update characteristic, where G^+ and G^- show a gradual increase and abrupt decrease. b) Bidirectional update characteristic, where G^+ and G^- show both gradual increase and gradual decrease. In both cases, the equivalent conductance $G = G^+ - G^-$ can be seen on the vertical axis, while G^+ and G^- are measured along the axis at +45° and −45°, respectively, with respect to the horizontal axis.

This type of unidirectional update is schematically reported in the diamond plot of **Figure** 7a, showing G^+ as a function of G^- on ±45° axis. In the diamond plot, the net value G is represented by the position along the vertical axis. For a unidirectional device, where G^+ and G^-can only increase, the position on the plot can only move toward the right along the G^+or G^-axis. As the position hits the boundary $G^+ = G_{max}$ or $G^- = G_{max}$, the conductances have to be shifted to smaller values along the horizontal axis, thus to preserve a constant value of the net G. Figure 7b shows instead the case of bidirectional update, such as the case of RRAM devices[116,121] or ionic transistors.[72–76] In this case, the position on the diamond plot can move in any direction; thus, resetting to a lower G is generally not necessary.

In general, the memory conductance does not only have a superior limit G_{max}, but also an inferior limit G_{min}, which possibly creates an additional constraint to net conductance G. This is schematically shown in Figure 6d, indicating a bidirectional update of G limited between G_{min} and G_{max}. In such a case, the differential synapse of Figure 5c is useful, as the zero conductance G=0 can be achieved by carefully tuning G^+ and G^- so that equal values are obtained to ensure the weight annihilation according to $G^+ - G^- = 0$. While this situation is straightforward with $G^+ = G^- = 0$, the presence of a minimum G might make the achievement of null G rather difficult.

Other sources of nonideality are the stochastic variation of conductance of Figure 6e, where an applied pulse can cause a relatively large, random increase, or decrease in conductance similar to Figure 4b. The weight-update granularity (i.e., the dynamic range of conductance is covered by only few individual increase/decrease steps) and stochasticity (i.e., the amplitude of each step is random) prevent the fine control of the weight, hence the network accuracy. A possible solution to large granularity as well as asymmetric weight update is the hybrid CMOS/PCM synapse of **Figure 8**. The hybrid synapse includes two differential synapses, one storing the most significant pair (MSP) whereas the other stores the least significant pair (LSP). Each element of the LSP synapse is a three-transistor, one-capacitor element for linear weight update, whereas the differential MSP synapse consists of two PCM memories with a 1T1R structure with nonvolatile storage.[122,123] In this way, the fine weight update is conducted in the highly linear capacitor with conductance g, which is then periodically aggregated to the PCM weight of conductance G. At each time, the conductance is given by $FG + g$, where F is a gain factor usually in the range of $F = 3$. This circuit structure allows to largely improve the accuracy of online training toward the one achieved by software offline training in a previous study.[122]

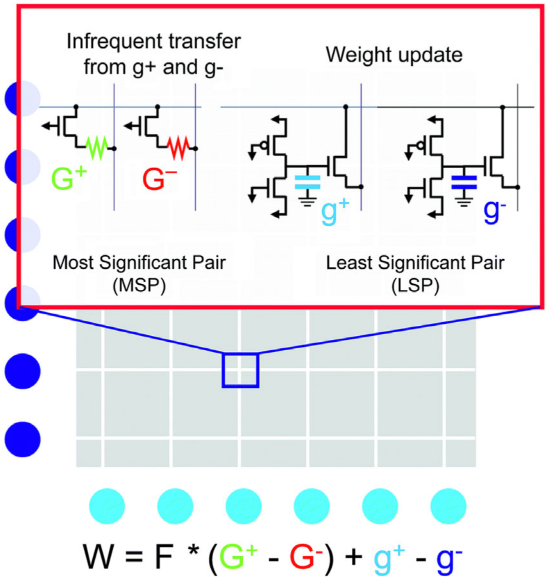
Infrequent transfer from g+ and g-
Weight update
G+
G–
g+
g-
Most Significant Pair (MSP)
Least Significant Pair (LSP)
W = F * (G+ - G-) + g+ - g-

Figure 8. Illustration of the hybrid CMOS/resistive synapse. The hybrid synapse includes a differential synapse with two three-transistor, one-capacitor element for linear weight update, combined with a differential synapse with 2 1T1R elements of PCM devices. The weight update term $g^+ - g^-$ contains the LSP in volatile memories, whereas the equivalent conductance $G^+ - G^-$ stores the MSP in nonvolatile memories. The total equivalent weight is given by $W = F{*}(G^+ - G^-) + g^+ - g^-$, where a gain $F = 3$ is usually assumed. Reproduced with permission.[123] Copyright 2019, RSC Publishing.

While the gradual update of the synaptic weight is generally beneficial for offline and online training, some memory devices show binary switching with abrupt increase and decrease in conductance, as shown in Figure 6f. This is the case for STT-MRAM, for instance, where magnetic polarization switches as a macrospin throughout the whole device area; thus, partial polarizations are generally not possible.[124] Similarly, some bipolar RRAM devices can display abrupt bidirectional switching.[125–127] In this case, the resulting neural network is inherently digital, which is referred to as the binarized neural network (BNN). Note that the gradual update of Figure 6a–e is not possible in BNNs, thus making online training particularly challenging. A stochastic version of online training can still be conducted in BNNs, utilizing RRAM devices where an internal state variable can be controlled by the application of voltage pulses.[125] Two synaptic weights can thus be associated with the RRAM device, namely, an internal, nonobservable weight W_{int} and an external, measurable W_{ext}. The internal weight maps the state variable of the device, e.g., the defect density and configuration within the filament region in **Figure 9**a, whereas W_{ext} is the device conductance which is

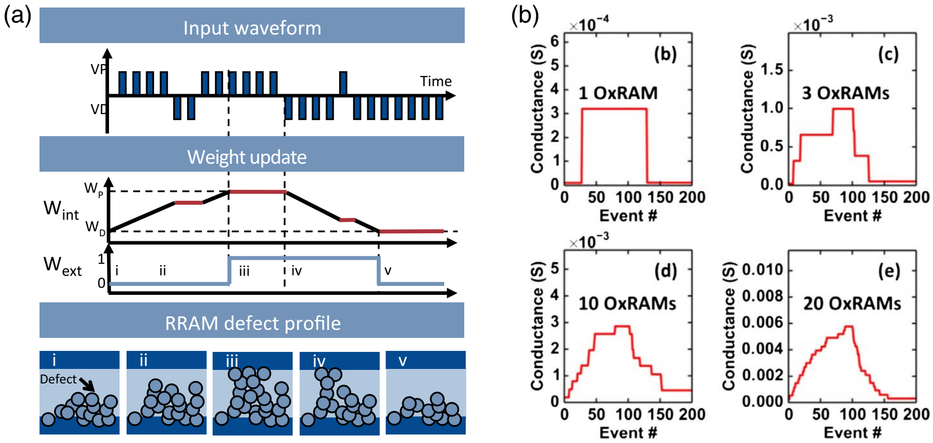

Figure 9. Illustration of stochastic synaptic memories with binary devices. a) Stochastic weight update, where positive/negative pulses (top) are applied, thus resulting in a gradual change of the internal weight and a binary update of the external weight W_{ext} (center). The defect configuration in the filamentary path is described by W_{int}, whereas the connection/disconnection of the filament to the TE/BE dictates the binary value of W_{ext}. Adapted with permission.[125] Copyright 2017, IEEE. b) Multidevice synapse, where the combination of the conductance of various binary memory devices can lead to an overall analogue synapse suitable for gradual weight update. Reproduced with permission.[127] Copyright 2015, IEEE.

high for the filament connecting the two electrodes, otherwise zero for all other configurations.[125] The application of pulses results in a continuous change of W_{int}, although W_{ext} will only change as W_{int} reaches a certain threshold. Note that the transition across the threshold is highly stochastic, as a certain W_{int} can correspond to various configuration of defects. The BNN can thus be trained with the backpropagation algorithm, similar to an analogue network.[125] In a similar approach, W_{ext} can be generated based on a measurable W_{int}, thus combining the benefits of the gradual update of the analogue weight and higher precision of the BNN.[126]

Another approach to online training with binary switching devices is the concept of multidevice synapse, where a single synapse including several binary devices in parallel effectively behaves as an analogue synapse.[127] Figure 9b shows simulation results for the update characteristics for increasing the number of memory elements. As the number of defects increases, the synapse update becomes increasingly analogue, thanks to the stochastic switching of individual elements.[127] In general, multidevice synapses also benefit from the better averaging of stochastic variations (Figure 6e), thus improving the weight controllability and the resulting network accuracy.[128]

As a final remark, the main advantages of online training of neural networks are 1) the energy efficiency, thanks to conducting the computation in the memory, thus taking advantage of inmemory MVM for forward propagation, and 2) the possibility of adapting the training to the specificity of the memory array, e.g., the presence of defects and device-to-device variations.[129] At the same time, online training for each individual neural network becomes energetically unfeasible; thus, the best approach is to conduct online training on a specific task on a master neural network, then transferring all synaptic weights to all other hardware samples. Techniques for defect-aware training have been proposed, e.g., by introducing random stuck short/open within the simulated network.[109]

5. IMC Circuit Nonidealities

Various nonidealities at the device levels, such as device variations, fluctuations, drift, and stuck open/short states, all affect the performance of the IMC circuit. For instance, the accuracy of the neural network, namely, the ability of recognizing objects or speech, might be degraded with respect to the ideal software accuracy for a certain set of synaptic weights. It has been shown that neural networks with a relatively large number of neurons for each layer display the highest resilience to variations, thanks to the better parallelism and the larger number of parameters to represent the data at each layer. On the other hand, relatively deep neural networks are instead more prone to device variations, due to the accumulation of errors during feed-forward propagation along the numerous layers of the deep neural network.[130]

In addition to device nonidealities, also, array parasitic can represent a serious concern for the IMC circuits. One of the major sources of circuit nonideality is the parasitic wire resistance in the array, causing current-resistance (IR) drop along the rows and columns of the memory array. This is shown in **Figure 10**a, where the wire resistance r between each cell is evidenced. Assuming a typical read voltage of 0.1 V, which is limited by noise, possible offsets of the voltage references and amplifiers, and possible mismatches in the CMOS periphery, and assuming an average device resistance $R = 100\,\text{k}\Omega$, each device is expected to carry an average current $I = 1\,\mu A$. Assuming the same current I for each device, then the overall voltage drop across the wire is $rI + 2rI + 3rI + \ldots + NrI = rIN^2/2$. For $N = 100$ and $r = 1\,\Omega$,[131] the voltage drop is around 5 mV, which is a significant contribution to the overall V_R. In addition to the large IR drop, the large total current NI also raises concerns in terms of power consumption, size of the decoder transistors, and of the SAs.

To reduce the line current and the corresponding IR drop, the average device resistance should be increased as much as possible, e.g., in the MΩ[115] or GΩ range.[65] A large device resistance,

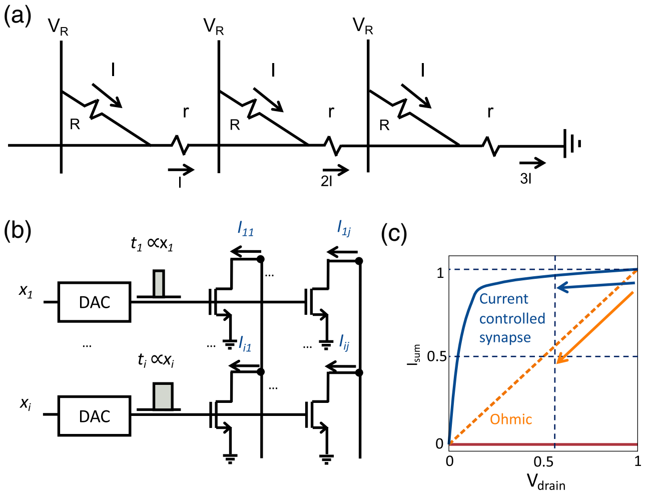

Figure 10. Parasitic voltage drop across array wires. a) Illustration of the voltage drop across a three-cell row with memory resistance R and inter-memory wire resistance r. b) Current-controlled synaptic element, where MVM is executed by multiplying the pulse width of the input signal with the saturated current of a synaptic transistor. c) Impact of IR drop for current-controlled synapses and ohmic devices. Adapted with permission.[65] Copyright 2019, IEEE.

however, might be more heavily impacted by resistance variations, fluctuations, and drift, which are generally most relevant for HRSs.[102,103] Also, offline training with programming/verifying such a large resistance also becomes challenging, due to the relatively long time needed to sense the extremely low read current within a very high resistance. Instead of increasing the memory resistance, one may also reduce the size of the individual memory arrays to reduce the overall IR drop. A tiled-RRAM architecture has been proposed to conveniently reduce the maximum array dimension of state-of-the-art RRAM devices with typical resistances.[132] However, reducing the array size also results in an increase in the number of the necessary analog–digital converters (ADCs), digital–analog converters (DACs), and other peripheral digital circuits, thus resulting in an overhead in terms of circuit area and power consumption.

Another solution to partially solve the issue of the IR drop is the current-controlled synaptic element of Figure 10b.[65] Here, a three-terminal device is considered, such as a FEFET, a Flash memory, or an ionic transistor, which serves as a current-controlled synapse operating in the saturated regime. The saturated current can be programmed by either online or offline training techniques and represents the synaptic weight, whereas the input information is encoded in the pulse width of the applied gate pulse. The synaptic currents are summed by Kirchhoff's law and used to discharge a pre-charged line or integrated on a capacitor. Note that this is an alternative way of conducting the MVM of Equation (1), where the pulse amplitude is replaced by the pulse width as input vector, the synapse conductance is replaced by the saturated current as weight matrix, and the summed current is replaced by the integrated charge as MVM output, according to

$$Q_i = \sum_j I_{ij} t_j \tag{4}$$

As shown in Figure 10c, the IR voltage drop plays a much smaller impact on the saturated characteristics of the synaptic transistors, compared with linear characteristics of two-terminal memory elements. This scheme has the additional advantages of digital input voltages at the gate, as well as the possibility of operating each transistor in the subthreshold regime, to enable low-current IMC.

6. IMC Circuit Architectures

Figure 11 shows various IMC architectures that have been developed to address application-specific computing problems. All architectures take advantage of the possibility of building compact memory arrays in a matrix shape and programming each memory device with an arbitrary analog value. The most popular architecture is the memory array for MVM acceleration in the analogue domain,[9–11] although other architectures can be built such as the content addressable memory (CAM)[133] and analogue IMC accelerators for solving inverse problems in one computing step.[104,134]

6.1. MVM Accelerators

Figure 11a shows a typical architecture for performing the MVM, namely $x = A \times \mathbf{b}$.[9–11] The input vector $\mathbf{b}$ is generally converted into the analog domain voltage vector $\mathbf{V}$ with a DAC; then, it is applied to crosspoint rows. The matrix A is mapped as conductance values of the memory elements in the crosspoint array. In principle, any of the cell structures of Figure 3, namely 1R, 1T1R, and 1S1R, can be used in the memory array. Array columns are connected to virtual ground such that the resulting current in each column is given by Equation (1) for a crosspoint of a given size N. Each current is converted into the voltage signal by a transimpedance amplifier (TIA), then converted into the digital domain by an ADC. This simple architecture can conduct MVM in one operational step with constant time independent of the size N of the problem, namely $O(1)$ time complexity.

MVM is the building block for accelerating neural networks, where sum of product must be executed many times during forward propagation. Here, vector $\mathbf{b}$ can be seen as the output neuron signal at a given layer, whereas the conductance matrix G maps the synaptic weights. IMC-based neural network accelerators have been widely demonstrated both for inference with offline supervised training[135,136] and for online training,[11,137] where MVM in the crosspoint array can be used to accelerate both the network evaluation and the training. Online training also allows to experience device nonidealities such as programming variations, limited window, and stuck open/short, thus resulting in a relatively high accuracy.[137] The nonlinear neuron activation is generally performed within the digital domain. The architecture is thus agnostic with respect to the type of training, which can span various learning algorithms such as supervised learning,[11,136,137] unsupervised learning,[138] and reinforcement learning.[139] Multilayer architectures, such as convolutional neural networks (CNN), can be accelerated within crosspoint memory arrays using separate arrays for each network layer,[122] arranging all networks in different locations within the same

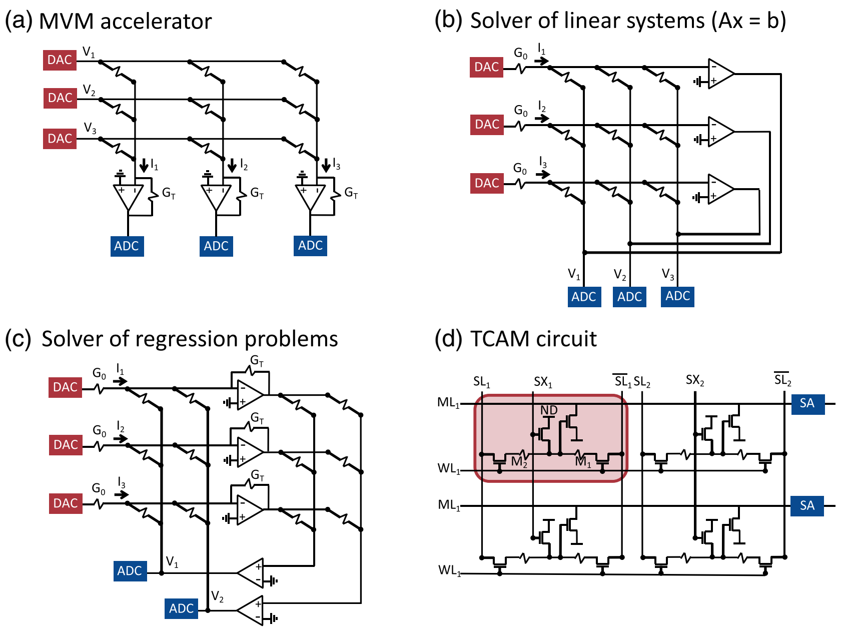

Figure 11. Array architectures for IMC. a) MVM accelerator including DAC at the input, TIAs for current–voltage conversion, and ADC at the output. b) TCAM array using terminal-resistive memory devices. c) IMC accelerator for solving a linear system $Ax = \mathbf{b}$. d) IMC accelerator for linear and logistic regression.

array[11] or even breaking each layer in several subarrays or tiles.[132] Integrated circuits comprising crosspoint memory arrays, DAC, TIA, and ADC have been already presented for neural network training,[140–143] showing software-equivalent accuracy and a performance density above 1 $\text{TOPs}^{-1}\,\text{mm}^{-2}$.[140]

The memory architecture in Figure 11a can be used to accelerate recurrent neural networks (RNNs) both for deep learning[137] and for the solution of constraint satisfaction problems (CSPs).[144–149] In the latter case, one can consider the MVM accelerator as a Hopfield-type RNN.[150,151] Hopfield RNNs are brain-inspired networks that can perform cognitive computing tasks on attractors, which are memory states that represent a minimum energy value in the landscape described by the network connectivity. Cognitive tasks in RNNs include attractor learning, attractor recall, and probabilistic model training.[152,153] When performing a recall operation, the Hopfield RNN converges to a stable state by minimizing the energy function $E = -\frac{1}{2}\sum G_{ij} V_i V_j$.[154] Thus, by programming the conductance matrix G with a function to optimize, the Hopfield RNN can iteratively find the minimum energy E.[150,151] However, many optimization problems have a nonconvex energy landscape, meaning that many local minima are present. As a result, a Hopfield RNN cannot solve the problem efficiently. This class of CSPs includes Max-SAT, Max-Cut, and the generic multidimensional expression of Sudoku.[155] To make the system capable of solving such nonconvex problems, computational annealing techniques are conducted, by introducing noise in the system, which is equivalent to increasing temperature in an annealing experiment. Simulated annealing allows the system to escape from local minima and reach the global minimum. The intrinsic noise in memory devices has been used as an experimental tool to accelerate computational annealing,[146,147] allowing for a speedup of the solution by a factor 30× compared with GPU[146] in a low-power RNN.

Analogue MVM can also be used to implement spiking neural networks (SNNs), which aim at mimicking the type of computation that takes place in the brain. In fact, while many SNNs have been developed based on standard CMOS technology,[5,6,156–158] it has been recognized that IMC allows for a more direct implementation of the neural network structure, as well as providing a better resemblance of the learning and spiking mechanisms of the brain. For instance, the biological learning rules, such as the STDP[159] and the Bienenstock–Cooper–Munro (BCM) rule for triplet-based learning,[160] can be naturally replicated in memory devices. For instance, STDP has been demonstrated in a relatively simple 1R structure,[161–163] one-transistor structures,[63,64] 1T1R structures,[164–166] and two-transistor/one-resistor (2T1R) structures.[167,168] The time-dependent dynamics of volatile RRAM[169,170] was also shown to feature bioinspired processes, such as STDP learning,[171] BCM learning,[172] short-term plasticity,[173] and oscillating neurons.[174,175] This type of neuromorphic, brain-inspired IMC is highly promising for ultralow-power smart sensors and biomedical devices interfacing with the brain, such as neuromorphic neuroprostheses.

Finally, analogue MVM in the memory can naturally accelerate algebraic computing problems such as image processing,[10] sparse coding,[176] and the solution of linear systems and differential equations.[177,178] In the latter case, numerical algorithms are adopted to break the algebraic problem in several iterative steps, including MVM within the memory architecture and a separate operation performed on a digital computer with floating point precision. This approach is referred to as mixed-precision computing, which takes advantage of the IMC concept only for accelerating the MVM operation.[178–180]

6.2. Analogue Computing Accelerators

Recently, it has been shown that the crosspoint array can be properly connected in a feedback loop with operational amplifiers (OAs) to solve a linear system of equation in one step without any iteration.[104] Figure 11c shows the circuit architecture for solving the linear system $Ax = \mathbf{b}$ in one step. The core architecture is the same as Figure 11a, namely a crosspoint architecture performing MVM between column voltages and conductance values G representing matrix A stored in the memory array. Vector $\mathbf{b}$ is applied as input analogue current $\mathbf{i}$, obtained as a DAC output signal applied to an input conductance G_0 connected to the virtual ground. Virtual ground is obtained at the input terminal of an OA where the output is connected to the array column with a feedback configuration. The OA generates an output voltage vector such that $G\mathbf{v} + \mathbf{i} = 0$, to support the MVM by Kirchoff's and Ohm's law. By rearranging this equation, one can obtain the unknown vector $\mathbf{v} = -G^{-1}\mathbf{i}$ in one step, which is the solution $x = \mathbf{v}$ to the linear system $Ax = \mathbf{b}$.

The circuit can be extended to matrices A which contain both positive and negative entries, by inverting the output voltage $\mathbf{v}$ of the OAs and applying it to a second crosspoint array G' parallel to G. As a result, one can solve a generic linear system $(B - C)x = \mathbf{b}$, where B and C are mapped in the crosspoint arrays G and G', respectively. As a special case, if the matrix G' is replaced by the diagonal matrix λI, where I is the identity matrix, and if the input vector is assumed $\mathbf{i} = 0$, then the problem reads $(A - \lambda I)x = 0$, where the unknown is the eigenvector of the matrix A. These linear algebra problems can be extended to differential equations, such as the Fourier equation or the Schrödinger equation in one step within a crosspoint array.[104] Note, however, that the circuit can only calculate the eigenvector for the maximum eigenvalue, which should be shown to allow for circuit implementation. This is the case, for instance, of the Pagerank, which is an algorithm for ranking webpages, where the maximum eigenvalue $\lambda = 1$ is always known.[181] To perform Pagerank, the matrix G of the connections between webpages is programmed into the memory array, and the eigenvector corresponding to the maximum eigenvalue is computed. Crosspoint circuits have been used to compute the Pagerank problem.[101,104]

While matrix A is always square in the circuit of Figure 11c, rectangular problems where the number of equations exceeds the number of unknowns can also be addressed with dedicated IMC architectures.[134] For instance, Figure 11d shows a double-feedback circuit to compute regression in one step. A current input vector $\mathbf{y}$ is applied as input current $\mathbf{i}$ by DAC connected to input conductance G_0. According to Kirchoff's law, the total current $G_X\mathbf{v} + \mathbf{i}$, where G_X is the conductance matrix of the left crosspoint array and $\mathbf{V}$ is the output voltage of the second stage of OAs, is converted to voltage $\mathbf{v}_\mathrm{R} = (G_X\mathbf{v} + \mathbf{i})/G_\mathrm{T}$ by the TIAs and applied to the right matrix. The right array encodes the same conductance G_X of the left array; thus, the output current is given by ${G_X}^\mathrm{T}(\mathbf{v}G_X + \mathbf{i})/G_\mathrm{T}$, which must be equal to zero due to the infinite input resistance of the second-stage OAs. As a result, the output voltage reads $\mathbf{v} = -({G_X}^\mathrm{T}G_X)^{-1}{G_X}^\mathrm{T}\mathbf{i}$, which represents the Moore–Penrose inverse $w = (X^\mathrm{T}X)^{-1}X^\mathrm{T}\mathbf{y}$, where matrix X is encoded into the conductance of the crosspoint array G_X. The solution is given in one step regardless of the matrix size, without any iteration.

The Moore–Penrose inverse can be used to compute the linear regression of a given set of data. By storing the independent variables X in the crosspoint array and applying the dependent variable y as the input current, the circuit output voltage is $\mathbf{v} = w$, which represents the linear coefficient of the best fitting line (or plane or hyperplane, depending on the number of dimensions).[134] The same concept can be extended to other types of regressions, such as polynomial regression and logistic regression. The latter can act as a building block for large-scale classification systems.[134]

The feedback configuration of the IMC circuits of Figure 11c,d allows for physical iteration in the analogue domain to find the solution of the problem with a relatively large size N. In principle, the solution time does not depend on the size N of the problem, thus resulting in $O(1)$ complexity. This low complexity makes IMC extremely promising for machine learning and other areas which rely on linear matrix computation. However, due to the nonidealities at the device level (e.g., device-to-device variations, drift, etc.) and circuit level (e.g., IR drop, etc.), it appears challenging for the IMC technology to reach the same precision as conventional digital circuits with floating-point precision. A more general study at the system level is still needed to meet these challenges and take full advantage of the low complexity and high energy efficiency of IMC.

6.3. Content Addressable Memories

Memory arrays are usually accessed by an address, which allows to select a certain memory bit to retrieve its content data. This operation is unambiguous, i.e., a single data bit corresponds to any specific address. However, many computing tasks require the opposite operation, namely searching the position, or multiple positions, where a given information is stored in the memory. This memory architecture, which is referred to as CAM, returns the data address in one clock cycle, independently from the memory size, thus allowing for an acceleration of data search with respect to software and other hardware approaches. CAM has been used to accelerate multiple computing tasks such as IP routing, image coding, and regular expression matching.[133] The conventional CMOS-based CAM requires a large area and complex circuit structure that limit its hardware implementation. On the other hand, CAM can be naturally implemented with IMC using two-terminal memory devices to allow for the significant increase in density.

Figure 11b shows a 2 × 2 ternary CAM (TCAM) array implemented with RRAM devices, where the single cell is highlighted. A TCAM is a more general type of CAM which is able to search not only binary values ("1" or "0") but also "don't care" value ("X").[182–186] Two operations can be performed on the TCAM array, namely writing and searching.[184] Signals $SX1$ and ND control the access transistor for write operation, whereas the selection transistor gate ($WL1$) is biased constantly at V_DD. To set the RRAM device on the right (M1), V_set is applied to $\overline{SL1}$, with $SL1$ kept at V_DD to turn off the left transistor, corresponding to device M2. The compliance current is regulated by the control voltage $SX1$ whereas ND is grounded. To reset the device M1,

$\overline{SL1}$ is grounded whereas ND is biased at V_{reset}. The same scheme can be applied to write M2 by inverting the signals $SL1$ and $\overline{SL1}$. State "0" corresponds to M1 in HRS and M2 in LRS, whereas state "1" corresponds to M1 in LRS and M2 in HRS. To write state "X," corresponding to "don't care," both M1 and M2 should be in HRS state. During search operation, the match line 1 ($ML1$) is precharged to V_{DD} whereas the search bit is applied at $SL1$. If there is match, then $ML1$ remains in the charged state. In fact, assuming that a "1" is searched whereas a "1" is stored in the cell, device M2 in HRS prevents discharge of $ML1$ to the grounded $\overline{SL1}$, thus maintaining the charged state of $ML1$. On the other hand, if there is no match, e.g., a "1" is searched whereas a "0" is stored in the cell, then device M2 in LRS connects $ML1$ to ground, thus inducing a fast discharge of the line. If state "X" is stored in the cell, then $ML1$ remains charged regardless of the input vector, as both M1 and M2 in HRS prevent connection of $ML1$ to $SL1$ and $\overline{SL1}$.

Thanks to its modular implementations, TCAM can be easily arranged in an array to search for large data patterns, as shown in Figure 11b. Giving an input word on $SL1$ and $SL2$, $ML1$ and $ML2$ remain charged only if each column data matches the input data, or data "X" are stored in the column. The $ML1$ and $ML2$ potential is rapidly probed by a SA to recognize a discharge if $ML1$ or $ML2$ drops below a certain threshold.

Interestingly, TCAM can be used to accelerate computing problems without the need for area/power consuming ADC and DAC and can be directly connected to the memory module such as DRAM.[186] The matchline (ML) discharge speed contains important information on the similarity between the search and stored value. For instance, if a weak LRS is written on M1, the discharge time while searching for "1" will be longer than the time corresponding to a full LRS. The difference between these two values can be translated into a Hamming distance and used to accelerate custom neural network training.[185] Moreover, analog-resistive CAM circuits have been proposed,[187] where the stored values represent a range and the ML will stay charged if the analog input signal is within the stored range. Analog CAM can be used as the inference machine for machine learning problems, such as decision trees and random forests.[187,188]

7. Conclusions

This work provides an overview on the devices, circuits, and architectures that enable data processing directly within the memory according to the so-called IMC paradigm. Emerging memory devices, including two-terminal and three-terminal devices, are first reviewed to clarify the operation principle and the associated advantages and disadvantages for computing. The device structures, including selector-free 1R, 1T1R, and 1S1R structures, have been discussed and compared. The most typical nonidealities of the memory concept are discussed with reference to different training processes, namely offline training consisting of memory programming operation and online training where the synaptic weights are updated in situ. Nonidealities at the array level are then considered, such as the IR drop along the array wires which dictates additional requirements for the memory resistance. Finally, the IMC architectures are reviewed with focus on MVM, TCAM, and analogue accelerators for solving linear algebra problems. Due to several advantages of performance, energy efficiency, and complexity, IMC appears extremely promising to accelerate many data-intensive computing tasks. Improvements in the device state control and resistance window are however needed to compensate the device nonidealities and improve the accuracy of IMC.

Acknowledgements

This article has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement no. 648635).

Conflict of Interest

The authors declare no conflict of interest.

Keywords

artificial intelligence, in-memory computing, machine learning, memories, neural networks

Received: March 14, 2020
Revised: April 13, 2020
Published online:

[1] G. Moore, *Electronics* **1965**, *38*, 8.
[2] M. Horowitz, in *2014 IEEE Int. Solid-State Circuits Conf. Digest of Technical Papers (ISSCC)*, IEEE, San Francisco, CA **2014**, pp. 10–14.
[3] K. Hai, *MIT Technology Review* 2019, June 6th.
[4] N. P. Jouppi, A. Borchers, R. Boyle, P. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, C. Young, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, N. Patil, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, et al., in *Proc. of the 44th Annual Int. Symp. on Computer Architecture – ISCA '17*, ACM Press, Toronto, ON, Canada **2017**, pp. 1–12.
[5] P. A. Merolla, J. V. Arthur, R. Alvarez Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, D. S. Modha, *Science* **2014**, *345*, 668.
[6] S.-C. Liu, G. Indiveri, *Proc. IEEE* **2015**, *103*, 1379.
[7] M. Zidan, J. P. Strachan, W. Lu, *Nat. Electron.* **2018**, *1*, 22.
[8] D. Ielmini, H.-S. Philip Wong, *Nat. Electron.* **2018**, *1*, 333.
[9] S. N. Truong, K.-S. Min, *J. Semicond. Technol. Sci.* **2014**, *14*, 356.
[10] 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.
[11] M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R. S. Williams, J. J. Yang, Q. Xia, J. P. Strachan, *Adv. Mater.* **2018**, *30*, 1705914.
[12] M. Cassinerio, N. Ciocchini, D. Ielmini, *Adv. Mater.* **2013**, *25*, 5975.
[13] S. Balatti, S. Ambrogio, Z. Wang, D. Ielmini, *IEEE J. Emerg. Sel. Topics Circuits Syst.* **2015**, *5*, 214.
[14] Z. Sun, E. Ambrosi, A. Bricalli, D. Ielmini, *Adv. Mater.* **2018**, *30*, 1802554.
[15] J. J. Yang, D. B. Strukov, D. R. Stewart, *Nat. Nanotechnol.* **2013**, *8*, 13.

[16] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, R. S. Williams, *Nature* **2010**, *464*, 873.
[17] S. Balatti, S. Ambrogio, D. Ielmini, *IEEE Trans. Electron. Devices* **2015**, *62*, 1831.
[18] P. Huang, J. Kang, Y. Zhao, S. Chen, R. Han, Z. Zhou, Z. Chen, W. Ma, M. Li, L. Liu, X. Liu, *Adv. Mater.* **2016**, *28*, 9758.
[19] B. Chen, F. Cai, J. Zhou, W. Ma, P. Sheridan, W. D. Lu, in *2015 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, Washington, DC **2015**, pp. 17.5.1–17.5.4.
[20] H.-S. Philip Wong, S. Salahuddin, *Nat. Nanotechnol.* **2015**, *10*, 191.
[21] D. Ielmini, S. Ambrogio, *Nanotechnology* **2019**, *31*, 092001.
[22] B. Govoreanu, G. S. Kar, Y.-Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. P. Radu, L. Goux, S. Clima, R. Degraeve, N. Jossart, O. Richard, T. Vandeweyer, K. Seo, P. Hendrickx, G. Pourtois, H. Bender, L. Altimime, D. J. Wouters, J. A. Kittl, M. Jurczak, in *2011 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, Washington, DC, USA **2011**, 31.36.31–31.36.34.
[23] C.-L. Tsai, F. Xiong, E. Pop, M. Shim, *ACS Nano* **2013**, *7*, 5360.
[24] S. Pi, C. Li, H. Jiang, W. Sia, H. Xin, J. J. Yang, Q. Xia, *Nat. Nanotechnol.* **2019**, *14*, 35.
[25] S. Yu, H.-Y. Chen, B. Gao, J. Kang, H.-S. P. Wong, *ACS Nano* **2013**, *7*, 2320.
[26] M. Yu, Y. Cai, Z. Wang, Y. Fang, Y. Liu, Z. Yu, Y. Pan, Z. Zang, J. Tan, X. Yang, M. Li, R. Huang, *Sci. Rep.* **2016**, *6*, 21020.
[27] R. Waser, M. Aono, *Nat. Mater.* **2007**, *6*, 833.
[28] H.-S. Philip Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, M.-J. Tsai, *Proc. IEEE* **2012**, *100*, 1951.
[29] D. Ielmini, *Semicond. Sci. Technol.* **2016**, *31*, 6.
[30] Z. Zhang, B. Gao, Z. Fang, X. Wang, Y. Tang, J. Sohn, H.-S. P. Wong, S. Wong, G.-Q. Lo, *IEEE Electron. Device Lett.* **2015**, *36*, 29.
[31] T. Hasegawa, K. Terabe, T. Tsuruoka, M. Aono, *Adv. Mater.* **2012**, *24*, 252.
[32] T. Sakamoto, H. Sunamura, H. Kawaura, T. Hasegawa, T. Nakayama, M. Aono, *Appl. Phys. Lett.* **2003**, *82*, 3033.
[33] U. Russo, D. Kamalanathan, D. Ielmini, A. L. Lacaita, M. N. Kozicki, *IEEE Trans. Electron Devices* **2009**, *56*, 5.
[34] F. Hui, E. Gustan-Gutierrez, S. Long, Q. Liu, A. K. Ott, A. C. Ferrari, M. Lanza, *Adv. Electron. Mater.* **2017**, *3*, 1600195.
[35] D. Ielmini, R. Bruchhaus, R. Waser, *Phase Transition* **2011**, *84*, 570.
[36] A. Sawa, *Mater. Today* **2008**, *11*, 28.
[37] C. W. Hsu, Y. F. Wang, C. C. Wan, I.-T. Wang, C.-T. Chou, W.-L. Lai, Y.-J. Lee, T.-H. Hou, *Nanotechnology* **2014**, *25*, 16.
[38] S. Raoux, W. Welnic, D. Ielmini, *Chem. Rev.* **2010**, *110*, 240.
[39] D. Ielmini, A. L. Lacaita, *Mater. Today* **2011**, *14*, 600.
[40] H.-S. P. Wong, S. Raoux, S.-B. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, K. E. Goodson, *Proc. IEEE* **2010**, *98*, 2201.
[41] G. Burr, *IBM J. Res. Dev.* **2008**, *52*, 465.
[42] T. Nirschl, J. B. Philipp, T. D. Happ, G. W. Burr, B. Rajendrant, M.-H. Lee, A. Schrott, M. Yang, M. Breitwischt, C.-F. Chen, E. Joseph, M. Lamorey, R. Chee, S.-H. Chen, S. Zaidi, S. Raoux, Y. C. Chen, Y. Zhu, R. Bergmann, H.-L. Lunge, C. Lam, in *2007 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, Washington, DC, USA **2007**, pp. 461–464.
[43] A. Athmanathan, M. Stanisavljevic, N. Papandreou, H. Pozidis, E. Eleftheriou, *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2016**, *6*, 1.
[44] G. Servalli, in *2009 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, Baltimore, MD, USA **2009**, pp. 1–4.
[45] F. Xiong, A. D. Liao, D. Estrada, E. Pop, *Science* **2011**, *332*, 568.
[46] J. Liang, R. G. D. Jeyasingh, H.-Y. Chen, H.-S. P. Wong, *IEEE Trans. Electron Devices* **2012**, *59*, 1155.
[47] D. Ielmini, A. L. Lacaita, D. Mantegazza, *IEEE Trans. Electron Devices* **2007**, *54*, 308.
[48] S. Kim, N. Sosa, M. BrightSky, D. Mori, W. Kim, Y. Zhu, K. Suu, C. Lam, in *2013 IEEE Int. Electron Devices Meeting*, IEEE, Washington, DC, USA **2013**, pp. 30.7.1–30.7.4.
[49] I. Giannopoulos, A. Sebastian, M. Le Gallo, V. P. Jonnalagadda, M. Sousa, M. N. Bonn, E. Eleftheriou, in *2018 IEEE Int. Electron Devices Meeting*, IEEE, San Francisco, CA, USA **2018**, pp. 27.7.1–27.7.4.
[50] C. Chappert, A. Fert, F. N. van Dau, *Nat. Mater.* **2007**, *6*, 813.
[51] B. N. Engel, J. Åkerman, B. Butcher, R. W. Dave, M. DeHerrera, M. Durlam, G. Grynkewich, J. Janesky, S. V. Pietambaram, N. D. Rizzo, J. M. Slaughter, K. Smith, J. J. Sun, S. Tehrani, *IEEE T-MAG* **2005**, *41*, 132.
[52] M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, M. Shoji, H. Hachino, C. Fukumoto, H. Nagao, H. Kano, in *2005 IEEE Int. Electron Devices Meeting*, IEEE, Washington, DC, USA **2005**, pp. 459–462.
[53] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. D. Gan, M. Endo, S. Kanai, J. Hayawaka, F. Matsukura, H. Ohno, *Nat. Mater.* **2010**, *9*, 721.
[54] S. Sakhare, M. Perumkunnil, T. Huynh Bao, S. Rao, W. Kim, D. Crotti, F. Yasin, S. Couet, J. Swerts, S. Kundu, D. Yakimets, R. Baert, H. R. Oh, A. Spessot, A. Mocuta, G. Sankar Kar, A. Furnemont, in *2018 IEEE Int. Electron Devices Meeting*, IEEE, San Francisco, CA, USA **2018**, 18.3.1–18.3-4.
[55] J. Grollier, D. Querlioz, M. D. Stiles, *Proc. IEEE* **2016**, *104*, 2024.
[56] T. Mikolajick, C. Dehm, W. Hartner, I. Kasko, M. J. Kastner, N. Nagel, M. Moert, C. Mazure, *Microelectron Reliab.* **2001**, *41*, 947.
[57] T. S. Böscke, J. Müller, D. Bräuhaus, U. Schröder, U. Böttger, in *IEDM Technical Digest*, IEEE, San Francisco, CA, USA **2011**, p. 547.
[58] A. Chanthbouala, A. Crassous, V. Garcia, K. Bouzehouane, S. Fusil, X. Moya, J. Allibe, B. Dlubak, J. Grollier, S. Xavier, C. Deranlot, A. Moshar, R. Proksch, N. D. Mathur, M. Bibes, A. Barthélémy, *Nat. Nanotechnol.* **2012**, *7*, 101.
[59] T.-Y. Wu, T.-S. Chang, H.-Y. Lee, S.-S. Sheu, W.-C. Lo, T.-H. Hou, H.-H. Huang, Y.-H. Chu, C.-C. Chang, M.-H. Wu, C.-H. Hsu, C.-T. Wu, M.-C. Wu, W.-W. Wu, in *2019 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA, USA **2019**, pp. 6.3.1–6.3.4.
[60] H. Mulaosmanovic, J. Ocker, S. Müller, U. Schroeder, J. Müller, P. Polakowski, S. Flachowsky, R. van Bentum, T. Mikolajick, S. Slesazeck, *ACS Appl. Mater. Interfaces* **2017**, *9*, 3792.
[61] R. Bez, E. Camerlenghi, A. Modelli, A. Visconti, *Proc. IEEE* **2003**, *91*, 489.
[62] F. Merrikh Bayat, X. Guo, M. Klachko, M. Prezioso, K. K. Likharev, D. B. Strukov, *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 4782.
[63] G. Malavena, M. Filippi, A. S. Spinelli, C. Monzio Compagnoni, *IEEE Trans. Electron Devices* **2019**, *66*, 4727.
[64] G. Malavena, M. Filippi, A. S. Spinelli, C. Monzio Compagnoni, *IEEE Trans. Electron Devices* **2019**, *66*, 4733.
[65] S. Cosemans, B. Verhoef, J. Doevenspeck, I. A. Papistas, F. Catthoor, P. Debacker, A. Mallik, D. Verkest, in *2019 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA, USA **2019**, pp. 22.2.1–22.2.4.
[66] T. Li, S. T. Hsu, B. D. Ulrich, L. Stecker, D. R. Evans, *Jpn. J. Appl. Phys.* **2002**, *41*, 6890.
[67] M. Tang, X. Xu, Z. Ye, Y. Sugiyama, H. Ishiwara, *IEEE Trans. Electron Devices* **2011**, *58*, 370.
[68] S. Sakai, M. Takahashi, K. Takeuchi, Q. H. Li, T. Horiuchi, S. Wang, K. Y. Yun, M. Takamiya, T. Sakurai, in *Proc. Non-Volatile Semiconductor Memory Workshop*, IEEE, Piscataway, NJ **2008**, p. 103.
[69] K. Florent, M. Pesic, A. Subirats, K. Banerjee, S. Lavizzari, A. Arreghini, L. Di Piazza, G. Potoms, F. Sebaai, S. R. C. McMitchell, M. Popovici, G. Groeseneken, J. Van Houdt, in *2018 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2018**, pp. 2.5.1–2.5.4.

[70] M. Cubukcu, O. Boulle, M. Drouard, K. Garello, C. Onur Avci, I. Mihai Miron, J. Langer, B. Ocker, P. Gambardella, G. Gaudin, *Appl. Phys. Lett.* **2014**, *104*, 042406.

[71] K. Garello, C. O. Avci, I. M. Miron, M. Baumgartner, A. Ghosh, S. Auffret, O. Boulle, G. Gaudin, P. Gambardella, *Appl. Phys. Lett.* **2014**, *105*, 212402.

[72] E. J. Fuller, F. E. Gabaly, F. Léonard, S. Agarwal, S. J. Plimpton, R. B. Jacobs-Gedrim, C. D. James, M. J. Marinella, A. A. Talin, *Adv. Mater.* **2017**, *29*, 1604310.

[73] Y. van de Burgt, E. Lubberman, E. J. Fuller, S. T. Keene, G. C. Faria, S. Agarwal, M. J. Marinella, A. Alec Talin, A. Salleo, *Nat. Mater.* **2017**, *16*, 414.

[74] J. Tang, D. Bishop, S. Kim, M. Copel, T. Gokmen, T. Todorov, S. Shin, K.-T. Lee, P. Solomon, K. Chan, W. Haensch, J. Rozen, in *2018 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2018**, pp. 13.1.1–13.1.4.

[75] E. J. Fuller, S. T. Keene, A. Melianas, Z. Wang, S. Agarwal, Y. Li, Y. Tuchman, C. D. James, M. J. Marinella, J. J. Yang, A. Salleo, A. A. Talin, *Science* **2019**, *364*, 570.

[76] S. Kim, J. A. Ott, T. Ando, H. Miyazoe, V. Narayanan, J. Rozen, T. Todorov, M. Onen, T. Gokmen, D. Bishop, P. Solomon, K.-T. Lee, M. Copel, D. B. Farmer, in *2019 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2019**, pp. 35.7.1–35.7.4.

[77] V. K. Sangwan, H.-S. Lee, H. Bergeron, I. Balla, M. E. Beck, K.-S. Chen, M. C. Hersam, *Nature* **2018**, *554*, 500.

[78] V. K. Sangwan, D. Jariwala, I. S. Kim, K.-S. Chen, T. J. Marks, L. J. Lauhon, M. C. Hersam, *Nat. Nanotechnol.* **2015**, *10*, 403.

[79] X. Zhu, D. Li, X. Liang, W. D. Lu, *Nat. Mater.* **2019**, *18*, 141.

[80] M.-C. Hsieh, Y.-C. Liao, Y.-W. Chin, C.-H. Lien, T.-S. Chang, Y.-D. Chih, S. Natarajan, M.-J. Tsai, Y.-C. King, C. J. Lin, in *2013 IEEE Int. Electron Devices Meeting*, IEEE, Washington, DC, USA **2013**, pp. 10.3.1–10.3.4.

[81] I. G. Baek, C. J. Park, H. Ju, D. J. Seong, H. S. Ahn, J. H. Kim, M. K. Yang, S. H. Song, E. M. Kim, S. O. Park, C. H. Park, C. W. Song, G. T. Jeong, S. Choi, H. K. Kang, C. Chung, in *2011 Int. Electron Devices Meeting*, IEEE, Washington, DC, USA **2011**, pp. 31.8.1–31.8.4.

[82] E. Linn, R. Rosezin, C. Kügeler, R. Waser, *Nat. Mater.* **2010**, *9*, 403.

[83] Y. C. Chen, C. F. Chen, C. T. Chen, J. Y. Yu, S. Wu, S. L. Lung, R. Liu, C. Y. Lu, in *2003 Int. Electron Devices Meeting*, IEEE, Washington, DC, USA **2003**, pp. 905–908.

[84] D. Ielmini, Y. Zhang, in *2006 Int. Electron Devices Meeting*, IEEE, San Francisco, CA, USA **2006**, pp. 1–4.

[85] L. Gao, P.-Y. Chen, R. Liu, S. Yu, *IEEE Trans. Electron Devices* **2016**, *63*, 3109.

[86] F. Li, X. Yang, A. T. Meeks, J. T. Shearer, K. Y. Le, *IEEE Trans. Device Mater. Reliab.* **2004**, *4*, 416.

[87] T.-Y. Liu, T. H. Yan, R. Scheuerlein, Y. Chen, J. K. Lee, G. Balakrishnan, G. Yee, H. Zhang, A. Yap, J. Ouyang, T. Sasaki, S. Addepalli, A. Al-Shamma, C.-Y. Chen, M. Gupta, G. Hilton, S. Joshi, A. Kathuria, V. Lai, D. Masiwal, M. Matsumoto, A. Nigam, A. Pai, J. Pakhale, C. H. Siau, X. Wu, R. Yin, L. Peng, J. Y. Kang, S. Huynh, H. Wang, N. Nagel, Y. Tanaka, M. Higashitani, T. Minvielle, C. Gorla, T. Tsukamoto, T. Yamaguchi, M. Okajima, T. Okamura, S. Takase, T. Hara, H. Inoue, L. Fasoli, M. Mofidi, R. Shrivastava, K. Quader, in *2013 IEEE Int. Solid-State Circuits Conference Digest of Technical Papers*, IEEE, San Francisco, CA **2013**, pp. 210–211.

[88] G. W. Burr, R. S. Shenoy, K. Virwani, P. Narayanan, A. Padilla, B. Kurdi, *J. Vac. Sci. Technol. B* **2014**, *32*, 040802.

[89] I. G. Baek, D. C. Kim, M. J. Lee, H.-J. Kim, E. K. Yim, M. S. Lee, J. E. Lee, S. E. Ahn, S. Seo, J. H. Lee, J. C. Park, Y. K. Cha, S. O. Park, H. S. Kim, I. K. Yoo, U.-I. Chung, J. T. Moon, B. I. Ryu, in *Int. Electron Devices Meeting, Technical Digest*, IEEE, Washington, DC, USA **2005**, p. 750.

[90] M.-J. Lee, Y. Park, B.-S. Kang, S.-E. Ahn, C. Lee, K. Kim, G. Stefanovich, J.-H. Lee, S.-J. Chung, Y.-H. Kim, C.-S. Lee, J. Bong, I.-G. Baek, I.-K. Yoo, in *2007 Int. Electron Devices Meeting*, IEEE, Washington, DC, USA **2007**, pp. 771–774.

[91] W. Lee, J. Park, J. Shin, J. Woo, S. Kim, G. Choi, S. Jung, S. Park, D. Lee, E. Cha, H. D. Lee, S. G. Kim, S. Chung, H. Hwang, in *2012 Symp. on VLSI Technology (VLSIT)*, IEEE, Honolulu, HI, USA **2012**, pp. 37–38.

[92] J. Woo, W. Lee, S. Park, S. Kim, D. Lee, G. Choi, E. Cha, in *2013 Symp. on VLSI Technology (VLSIT)*, IEEE, Kyoto, Japan **2013**, pp. 168–169.

[93] M. Son, J. Lee, J. Park, J. Shin, G. Choi, S. Jung, W. Lee, S. Kim, S. Park, H. Hwang, *IEEE Electron Device Lett.* **2011**, *32*, 1579.

[94] K. Gopalakrishnan, R. S. Shenoy, C. T. Rettner, K. Virwani, D. S. Bethune, R. M. Shelby, G. W. Burr, A. Kellock, R. S. King, K. Nguyen, A. N. Bowers, M. Jurich, B. Jackson, A. M. Friz, T. Topuria, P. M. Rice, B. N. Kurdi, in *2010 Symp. on VLSI Technology*, IEEE, Honolulu, HI **2010**, pp. 205–206.

[95] D. C. Kau, S. Tang, I. V. Karpov, R. Dodge, B. Klehn, J. A. Kalb, J. Strand, A. Diaz, N. Leung, J. Wu, Sean Lee, T. Langtry, K. W. Chang, C. Papagianni, J. Lee, J. Hirst, S. Erra, E. Flores, N. Righos, H. Castro, G. Spadini, in *2009 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, Baltimore, MD, USA **2009**, pp. 1–4.

[96] M.-J. Lee, D. Lee, S.-H. Cho, J.-H. Hur, S.-M. Lee, D. H. Seo, D.-S. Kim, M.-S. Yang, S. Lee, E. Hwang, M. R. Uddin, H. Kim, U.-I. Chung, Y. Park, I.-K. Yoo, *Nat. Commun.* **2013**, *4*, 2629.

[97] T. Kim, H. Choi, M. Kim, J. Yi, D. Kim, S. Cho, H. Lee, C. Hwang, E.-R. Hwang, J. Song, S. Chae, Y. Chun, J.-K. Kim, in *2018 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2018**, pp. 37.1.1–37.1.4.

[98] H. Y. Cheng, W. C. Chien, I. T. Kuo, C. W. Yeh, L. Gignac, W. Kim, E. K. Lai, Y. F. Lin, R. L. Bruce, C. Lavoie, C. W. Cheng, A. Ray, F. M. Lee, F. Carta, C. H. Yang, M. H. Lee, H. Y. Ho, M. BrightSky, H. L. Lung, in *2018 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2018**, pp. 37.3.1–37.3.4.

[99] D. Ielmini, *IEEE Trans. Electron Devices* **2011**, *58*, 4309.

[100] V. Milo, C. Zambelli, P. Olivo, E. Pérez, M. K. Mahadevaiah, O. G. Ossorio, Ch. Wenger, D. Ielmini, *APL Mater.* **2019**, *7*, 081120.

[101] Z. Sun, E. Ambrosi, G. Pedretti, A. Bricalli, D. Ielmini, *IEEE Trans. Electron Devices* **2020**, *67*, 1466.

[102] S. Balatti, S. Ambrogio, D. C. Gilmer, D. Ielmini, *IEEE Electron Device Lett.* **2013**, *34*, 861.

[103] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, D. Ielmini, *IEEE Trans. Electron Devices* **2014**, *61*, 2912.

[104] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, W. Wang, D. Ielmini, *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 4123.

[105] Y.-H. Lin, C.-H. Wang, M.-H. Lee, D.-Y. Lee, Y.-Y. Lin, F.-M. Lee, H.-L. Lung, K.-C. Wang, T.-Y. Tseng, C.-Y. Lu, *IEEE Trans. Electron Devices* **2019**, *66*, 1289.

[106] S. Balatti, S. Ambrogio, Z. Wang, S. Sills, A. Calderoni, N. Ramaswamy, D. Ielmini, *IEEE Trans. Electron Devices* **2015**, *62*, 3365.

[107] R. Carboni, S. Ambrogio, W. Chen, M. Siddik, J. Harms, A. Lyle, W. Kula, G. Sandhu, D. Ielmini, *IEEE Trans. Electron Devices* **2018**, *65*, 2470.

[108] M. Zhao, H. Wu, B. Gao, X. Sun, Y. Liu, P. Yao, Y. Xi, X. Li, Q. Zhang, K. Wang, S. Yu, H. Qian, in *2018 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2018**, pp. 20.2.1–20.2.4.

[109] T. Gokmen, M. J. Rasch, W. Haensch, in *2019 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA, USA **2019**, pp. 22.3.1–22.3.4.
[110] S. Ambrogio, S. Balatti, V. McCaffrey, D. C. Wang, D. Ielmini, *IEEE Trans. Electron Devices* **2015**, *62*, 3812.
[111] N. Ciocchini, M. Cassinerio, D. Fugazza, D. Ielmini, *IEEE Trans. Electron Devices* **2012**, *59*, 3084.
[112] D. Ielmini, D. Sharma, S. Lavizzari, A. L. Lacaita, *IEEE Trans. Electron Devices* **2009**, *56*, 1070.
[113] S. Gabardi, S. Caravati, G. C. Sosso, J. Behler, M. Bernasconi, *Phys. Rev. B* **2015**, *92*, 054201.
[114] P. Fantini, S. Brazzelli, E. Cazzini, A. Mani, *Appl. Phys. Lett.* **2012**, *100*, 013505.
[115] T. Gokmen, Y. Vlasov, *Front. Neurosci.* **2016**, *10*, 333.
[116] S. Yu, *Proc. IEEE* **2018**, *106*, 260.
[117] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, Y. Xie, in *2016 ACM/IEEE 43rd Annual Int. Symp. on Computer Architecture (ISCA)*, IEEE, Seoul, South Korea **2016**, pp. 27–39.
[118] G. W. Burr, R. M. Shelby, S. Sidler, C. di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, H. Hwang, *IEEE Trans. Electron Devices* **2015**, *62*, 3498.
[119] Y. LeCun, Y. Bengio, G. Hinton, *Nature* **2015**, *521*, 436.
[120] H. Kim, M. Rasch, T. Gokmen, T. Ando, H. Miyazoe, J.-J. Kim, J. Rozen, S. Kim, *ArXiv preprint*, **2019**, arXiv:1907.10228.
[121] J.-W. Jang, S. Park, G. W. Burr, H. Hwang, Y.-H. Jeong, *IEEE Electron Device Lett.* **2015**, *36*, 457.
[122] S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat, C. di Nolfo, S. Sidler, M. Giordano, M. Bodini, N. C. P. Farinha, B. Killeen, C. Cheng, Y. Jaoudi, G. W. Burr, *Nature* **2018**, *558*, 60.
[123] L. P. Romero, S. Ambrogio, M. Giordano, G. Cristiano, M. Bodini, P. Narayanan, H. Tsai, R. M. Shelby, G. W. Burr, *Faraday Discuss.* **2019**, *213*, 371.
[124] R. Carboni, E. Vernocchi, M. Siddik, J. Harms, A. Lyle, G. Sandhu, D. Ielmini, *IEEE Trans. Electron Devices* **2019**, *66*, 4176.
[125] C. Chang, J. Liu, Y. Shen, T. Chou, P. Chen, I. Wang, C. Su, M. Wu, B. Hudec, C. Chang, C. Tsai, T. Chang, H.-S. P. Wong, T. Hou, in *2017 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2017**, pp. 11.6.1–11.6.4.
[126] Z. Zhou, P. Huang, Y. C. Xiang, W. S. Shen, Y. D. Zhao, Y. L. Feng, B. Gao, H. Q. Wu, H. Qian, L. F. Liu, X. Zhang, X. Y. Liu, J. F. Kang, in *2018 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2018**, pp. 20.7.1–20.7.4.
[127] D. Garbin, E. Vianello, O. Bichler, Q. Rafhay, C. Gamrat, G. Ghibaudo, B. DeSalvo, L. Perniola, *IEEE Trans. Electron Devices* **2015**, *62*, 2494.
[128] I. Boybat, M. Le Gallo, S. R. Nandakumar, T. Moraitis, T. Parnell, T. Tuma, B. Rajendran, Y. Leblebici, A. Sebastian, E. Eleftheriou, *Nat. Commun.* **2018**, *9*, 2514.
[129] M. Rao, Z. Wang, C. Li, H. Jiang, R. Midya, P. Lin, D. Belkin, W. Song, S. Asapu, Q. Xia, J. J. Yang, in *2019 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2019**, pp. 35.4.1–35.4.4.
[130] T.-J. Yang, V. Sze, in *2019 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA, USA **2019**, pp. 22.1.1–22.1.4.
[131] International Technology Roadmap for Semiconductors (ITRS), Available at www.itrs2.net/2013-itrs.html
[132] Q. Wang, X. Wang, S. H. Lee, F.-H. Meng, W. D. Lu, in *2019 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA, USA **2019**, pp. 14.4.1–14.4.4.
[133] K. Pagiamtzis, A. Sheikholeslami, *IEEE J. Solid-State Circuits* **2006**, *41*, 712.
[134] Z. Sun, G. Pedretti, A. Bricalli, D. Ielmini, *Sci. Adv.* **2020**, *6*, eaay2378.
[135] V. Milo, C. Zambelli, P. Olivo, E. Perez, O. G. Ossorio, Ch. Wenger, D. Ielmini, in *ESSDERC 2019 – 49th European Solid-State Device Research Conference (ESSDERC)*, IEEE, Cracow, Poland **2019**, pp. 174–177.
[136] P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S. P. Wong, H. Qian, *Nat. Commun.* **2017**, *8*, 15199.
[137] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, Q. Xia, *Nat. Commun.* **2018**, *9*, 2385.
[138] S. Oh, Y. Shi, X. Liu, J. Song, D. Kuzum, *IEEE Electron Device Lett.* **2018**, *39*, 1768.
[139] Z. Wang, C. Li, P. Lin, M. Rao, Y. Nie, W. Song, Q. Qiu, Y. Li, P. Yan, J. P. Strachan, N. Ge, N. McDonald, Q. Wu, M. Hu, H. Wu, R. S. Williams, Q. Xia, J. J. Yang, *Nat. Mach. Intell.* **2019**, *1*, 434.
[140] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, H. Qian, *Nature* **2020**, *577*, 641.
[141] W. Chen, C. Dou, K. Li, W.-Y. Lin, P.-Y. Li, J.-H. Huang, J.-H. Wang, W.-C. Wei, C.-X. Xue, Y.-C. Chiu, Y.-C. Kinh, C.-J. Lin, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, J. J. Yang, M.-S. Ho, M.-F. Chang, *Nat. Electron* **2019**, *2*, 420.
[142] F. Cai, J. M. Correll, S. H. Lee, Y. Lim, V. Bothra, Z. Zhang, M. P. Flynn, W. D. Lu, *Nat. Electron* **2019**, *2*, 290.
[143] S. Yin, X. Sun, S. Yu, J. Seo, *ArXiv preprint*, **2019**, arXiv:1909.07514.
[144] S. Kumar, J. P. Strachan, R. S. Williams, *Nature* **2017**, *548*, 318.
[145] J. H. Shin, Y. J. Jeong, M. A. Zidan, Q. Wang, W. D. Lu, in *2018 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2018**, pp. 3.3.1–3.3.4.
[146] F. Cai, S. Kumar, T. V. Vaerenbergh, R. Liu, C. Li, S. Yu, Q. Xia, J. J. Yang, R. Beausoleil, W. Lu, J. P. Strachan, *ArXiv preprint*, **2019**, arXiv:1903.11194.
[147] M. R. Mahmoodi, H. Kim, Z. Fahimi, H. Nili, L. Sedov, V. Polishchuk, D. B. Strukov, in *2019 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2019**, pp. 14.7.1–14.7.4.
[148] M. R. Mahmoodi, M. Prezioso, D. B. Strukov, *Nat. Commun.* **2019**, *10*, 5113.
[149] M. N. Bojnordi, E. Ipek, in *2016 IEEE Int. Symp. on High Performance Computer Architecture (HPCA)*, IEEE, Barcelona, Spain **2016**, pp. 1–13.
[150] J. J. Hopfield, D. W. Tank, *Science* **1986**, *233*, 625.
[151] J. J. Hopfield, D. W. Tank, *Biol. Cybern*, **1985**, *52*, 141.
[152] V. Milo, D. Ielmini, E. Chicca, in *2017 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA, USA **2017**, pp. 11.2.1–11.2.4.
[153] Y. Zhou, H. Wu, B. Gao, W. Wu, Y. Xi, P. Yao, S. Zhang, Q. Zhang, H. Qian, *Adv. Funct. Mater.* **2019**, *29*, 1900155.
[154] J. J. Hopfield, *Proc. Natl. Acad. Sci.* **1982**, *79*, 2554.
[155] M. R. Garey, *Computers and Intractability*, Freeman, New York **1979**.
[156] S. B. Furber, F. Galluppi, S. Temple, L. A. Plana, *Proc. IEEE* **2014**, *102*, 652.
[157] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. Civit Ballcels, T. Serrano-Gotarredona, A. J. Acosta-Jimenez, B. Linares-Barranco, *IEEE Trans. Neural Networks* **2009**, *20*, 1417.
[158] E. Chicca, F. Stefanini, C. Bartolozzi, G. Indiveri, *Proc. IEEE* **2014**, *102*, 1367.
[159] G.-Q. Bi, M.-M. Poo, *J. Neurosci.* **1998**, *18*, 10464.
[160] E. L. Bienenstock, L. N. Cooper, P. W. Munro, *J. Neurosci.* **1982**, *2*, 32.
[161] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, W. Lu, *Nano Lett.* **2010**, *10*, 1297.
[162] C. Zamarreño-Ramos, L. A. Camuñas-Mesa, J. A. Pérez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, B. Linares-Barranco, *Front. Neurosci.* **2011**, *5*, 26.

[163] D. Kuzum, R. G. D. Jeyasingh, B. Lee, H.-S. P. Wong, *Nano Lett.* **2012**, *12*, 2179.

[164] S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, N. Ramaswamy, D. Ielmini, *IEEE Trans. Electron Devices* **2016**, *63*, 1508.

[165] G. Pedretti, V. Milo, S. Ambrogio, R. Carboni, S. Bianchi, A. Calderoni, N. Ramaswamy, A. S. Spinelli, D. Ielmini, *Sci. Rep.* **2017**, *7*, 5288.

[166] Z.-Q. Wang, S. Ambrogio, S. Balatti, D. Ielmini, *Front. Neurosci.* **2015**, *8*, 438.

[167] S. Kim, M. Ishii, S. Lewis, T. Perri, M. BrightSky, W. Kim, R. Jordan, G. W. Burr, N. Sosa, A. Ray, J.-P. Han, C. Miller, K. Hosokawa, C. Lam, in *IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA, USA **2015**, p. 1.

[168] A. Bricalli, E. Ambrosi, M. Laudato, M. Maestro, R. Rodriguez, D. Ielmini, *IEEE Trans. Electron Devices* **2018**, *65*, 122.

[169] R. Midya, Z. Wang, J. Zhang, S. E. Savel'ev, C. Li, M. Rao, M. H. Jang. S. Joshi, H. Jiang, P. Lin, K. Norris, N. Ge, Q. Wu, M. Barnell, Z. Li, H. L. Xin, R. S. Williams, Q. Xia, J. J. Yang, *Adv. Mater.* **2017**, *29*, 1604457.

[170] Z. Wang, S. Joshi, S. E. Savel'ev, H. Jiang, R. Midya, P. Lin, M. Hu, N. Ge, J. P. Strachan, Z. Li, Q. Wu, M. Barnell, G.-L. Li, H. L. Xin, R. S. Williams, Q. Xia, J. J. Yang, *Nat. Mater.* **2017**, *16*, 101.

[171] Z. Wang, T. Zeng, Y. Ren, Y. Lin, H. Xu, X. Zhao, Y. Liu, D. Ielmini, *Nat. Commun.* **2020**, *11*, 1510.

[172] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J. K. Gimzewski, M. Aono, *Nat. Mater.* **2011**, *10*, 591.

[173] M. D. Pickett, G. Medeiros-Ribeiro, R. S. Williams, *Nat. Mater.* **2013**, *12*, 114.

[174] Q. Hua, H. Wu, B. Gao, Q. Zhang, W. Wu, Y. Li, X. Wang, W. Hu, H. Qian, *Global Challenges* **2019**, *3*, 1900015.

[175] Z. Wang, S. Joshi, S. Savel'ev, W. Song, R. Midya, Y. Li, M. Rao, P. Yan, S. Asapu, Y. Zhuo, H. Jiang, P. Lin, C. Li, J. H. Yoon, N. K. Upadhyay, J. Zhang, M. Hu, J. P. Strachan, M. Barnell, Q. Wu, H. Wu, R. S. Williams, Q. Xia, J. J. Yang, *Nat. Electronics* **2018**, *1*, 137.

[176] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, W. D. Lu, *Nat. Nanotechnol.* **2017**, *12*, 784.

[177] M. A. Zidan, Y. Jeong, J. Lee, B. Chen, S. Huang, M. J. Kushner, W. D. Lu, *Nat. Electron.* **2018**, *1*, 411.

[178] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, E. Eleftheriou, *Nat. Electron.* **2018**, *1*, 246.

[179] M. Le Gallo, A. Sebastian, G. Cherubini, H. Giefers, E. Eleftheriou, *IEEE Trans. Electron Devices* **2018**, *65*, 4304.

[180] I. Richter, K. Pas, X. Guo, R. Patel, J. Liu, E. Ipek, E. G. Friedman, in *Government Microcircuit Applications & Critical Technology Conf.*, GOMAC, St. Louis, MO **2015**.

[181] K. Bryan, T. Leise, *SIAM Rev.* **2006**, *48*, 569.

[182] C. E. Graves, S.-T. Lam, X. Li, L. Kiyama, M. Foltin, M. P. Hardy, J. P. Strachan, C. Li, X. Sheng, W. Ma, S. R. Chalamalasetti, D. Miller, J. S. Ignowski, B. Buchanan, L. Zheng, *IEEE Trans. Nanotechnol.* **2019**, *18*, 963.

[183] Q. Guo, X. Guo, Y. Bai, E. İpek, in *Proc. of the 44th Annual IEEE/ACM Int. Symp. on Microarchitecture – MICRO-44 '11*, ACM Press, Porto Alegre, Brazil **2011**, p. 339.

[184] L.-Y. Huang, M.-F. Chang, C. H. Chuang, C.-C. Kuo, C.-F. Chen, G.-H. Yang, H.-J. Tsai, T.-F. Chen, S.-S. Sheu, K.-L. Su, F. T. Chen, T.-K. Ku, M.-J. Tsai, M.-J. Kao, in *2014 Symp. on VLSI Circuits Digest of Technical Papers*, IEEE, Honolulu, HI, USA **2014**, pp. 1–2.

[185] K. Ni, X. Yin, A. F. Laguna, S. Joshi, S. Dünkel, M. Trentzsch, J. Müller, S. Beyer, M. Niemier, X. S. Hu, S. Datta, *Nat. Electron.* **2019**, *2*, 521.

[186] Q. Guo, X. Guo, R. Patel, E. G. Friedman, in *2013 ACM/IEEE 43rd Annual Int. Symp. on Computer Architecture (ISCA)*, IEEE, Tel-Aviv, Israel, **2013**, pp. 189–200.

[187] C. Li, C. E. Graves, D. Miller, J. P. Strachan, *Nat. Commun.* **2020**, *11*, 1638.

[188] T. Tracy, Y. Fu, I. Roy, E. Jonas, P. Glendenning, in *High Performance Computing* (Eds: J. M. Kunkel, P. Balaji, J. Dongarra), Springer International Publishing, Cham **2016**, pp. 200–218.