SCIENCE CHINA
Information Sciences

• REVIEW •

June 2018, Vol. 61 060422:1–060422:9
https://doi.org/10.1007/s11432-017-9424-y

Special Focus on Memristive Devices and Neuromorphic Computing

Neuromorphic computing with memristive devices

Wen MA, Mohammed A. ZIDAN & Wei D. LU*

Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor 48109*, USA*

Received 7 December 2017/Accepted 26 February 2018/Published online 15 May 2018

Abstract Technology advances in the last a few decades have resulted in profound changes in our society, from workplaces to living rooms to how we socialize with each other. These changes in turn drive further technology developments, as the exponential growth of data demands ever increasing computing power. However, improvements in computing capacity from device scaling alone is no longer sufficient, and new materials, devices, and architectures likely need to be developed collaboratively to meet present and future computing needs. Specifically, devices that offer co-located memory and computing characteristics, as represented by memristor devices and memristor-based computing systems, have attracted broad interest in the last decade. Besides tremendous appeal in data storage applications, memristors offer the potential for efficient hardware realization of neuromorphic computing architectures that can effectively address the memory and energy walls faced by conventional von Neumann computing architectures. In this review, we evaluate the state-of-the-art in memristor devices and systems, and highlight the potential and challenges of applying such devices and architectures in neuromorphic computing applications. New directions that can lead to general, efficient in-memory computing systems will also be discussed.

Keywords memristor, resistive random-access-memory (RRAM), neuromorphic computing, non-von Neumann, process in-memory

Citation Ma W, Zidan M A, Lu W D. Neuromorphic computing with memristive devices. Sci China Inf Sci, 2018, 61(6): 060422, https://doi.org/10.1007/s11432-017-9424-y

1 Introduction

Memristors (memristive devices) have gained broad interest recently as a promising candidate for future data storage and efficient computing systems [1–5]. At the device level, memristor-based memory offers high-density and fast on-chip storage and can help extend the lifetime of classical computing architectures based on Boolean logic [2,6]. At the system level, a new class of analog/digital neuromorphic architectures have been developed [5,7–9], which can exploit the native physical properties of these resistive switching (RS) devices to directly and naturally implement brain-inspired computing paradigms [1,10–12], making memristor-based devices and systems highly attractive for efficiently processing data-intensive tasks at very low power in both the near term and the long term.

Memristors are by definition two-terminal systems that show RS effects. From a material and device point of view, the RS effect can be attributed to different switching mechanisms [13], as summarized in Figure 1. In particular, redox-based memristors are extensively studied, and are often called resistive random-access-memory (RRAM) for memory applications [14,15]. These devices are generally simple in structure and nanoscale in dimensions (scaling < 10 nm has been demonstrated [16]), while at the same time offering excellent performance in terms of switching speed [17], retention, and write/erase

* Corresponding author (email: wluee@eecs.umich.edu)

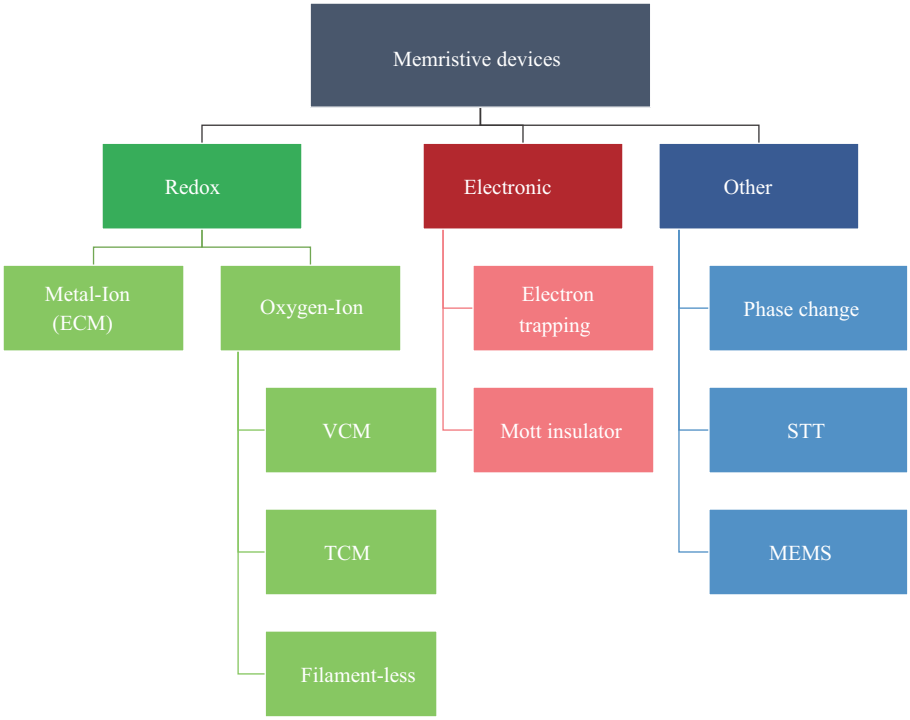

Figure 1 (Color online) Categories of memristive devices based on the RS mechanism [13] @ Copyright 2017 Springer Nature.

cycling [18]. In a typical device, the resistance state can be reversely switched from a high-resistance state (HRS) to a low-resistance state (LRS) during the SET process, when the bias voltage is above the SET threshold voltage. The device will maintain the new resistance value, thus can store the data in a non-volatile fashion, until it goes through a RESET process where the resistance can be switched back to the HRS, and vice versa (Figure 2). In general, RRAM devices are fabricated in a metal-insulator-metal (MIM) structure, with two metal electrodes sandwiching a thin film dielectric layer. Due to its nanoscale thickness, the dielectric layer can act as a solid electrolyte that accommodates cation or anion re-distribution within it [19], since at these dimensions even moderate voltage drops can create significant electric fields that exponentially speed up the ionic oxidation, migration and reduction processes. These processes in turn lead to the creation or annihilation of a conductive filament in the switching layer, and consequently the RS effects.

Other non-redox based RS systems can be driven by electronic effects such as electron trapping in an insulator [20] and insulator-metal transition effects in a Mott material [21]. Phase change memory (PCM) devices can also be included in the broad category of memristive systems, and are based on switching between the amorphous (high resistance) phase or crystalline (low resistance) phase of the material [22]. Spin transfer torque (STT) memory devices rely on the switching of the relative magnetic orientation of in spin valve structure, with parallel orientation of the two magnetic layers leading to LRS and antiparallel orientation leading to HRS [23]. In this review paper we will focus on the redox-based memristor devices as these devices have now been extensively studied in memory and neuromorphic computing applications.

Besides their potential in high-density, high-performance non-volatile memory applications, perhaps the most intriguing aspect of memristive devices is their application in emerging computing architectures. These architectures offer the potential to process large amounts of data efficiently, to satisfy the challenges presented by current and future computing needs. While some of the neural network and computational concepts are not necessarily new, recent advances in hardware, including memristor-based devices and circuits, made it possible to implement efficient hardware for edge processing and even server-like applications. The new device properties also enabled the development of new architectures that can take advantage of the high parallelism and co-located memory and compute properties offered by the devices,

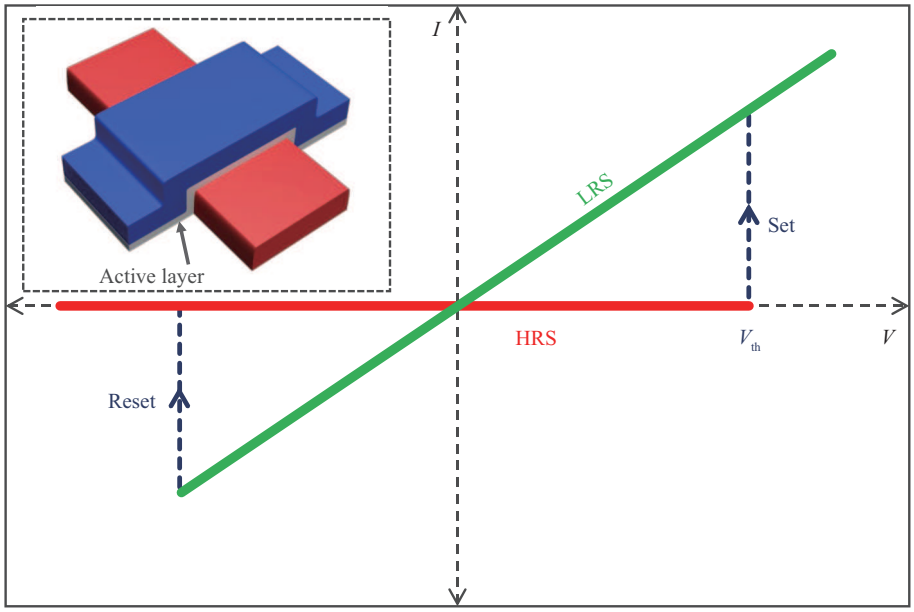

Figure 2 (Color online) Bipolar switching behavior of an RRAM device. (Inset) Crosspoint structure that is typically used to implement the two-terminal RRAM devices and arrays [13] @ Copyright 2017 Springer Nature.

and helped drive new architecture and algorithm developments.

We will evaluate the architecture and hardware developments, such as neuromorphic computing and biologically inspired computing approaches, driven by recent device discoveries. These neuromorphic computing systems, and in-memory computing systems in general, can overcome the von-Neumann bottleneck and are of particular relevance to tasks that are of high interest today. We will start from device-level developments and requirements and move on to system-level implementations and possible new research directions enabled by these fascinating devices.

2 Memristor as synapse

The quasi-continuous tuning of the device conductance with a memory effect, as shown in Figure 3, is a key enabling factor of memristor-based neuromorphic circuits and architectures. Generally speaking, the conductance in memristors plays the role analogous to synaptic weight in biological systems. Similar to a biological synapse, a memristor device can be used to connect a pair of computing elements (artificial neurons) and the (tunable) memristor conductance directly controls the connectivity strength between the neurons [24]. These connectivity patterns among neurons provide both memory (as values represented by the weights) and compute (since the connectivity strength directly regulates signal propagation) functions of the system. Additionally, the connectivity can be adjusted (trained) on-line by directly modifying the memristor conductance values with voltage pulses (analogous to spikes). These properties make it very attractive to use memristors to build neuromorphic systems, e.g., artificial neural networks with input and output elements (artificial neurons) connected by memristors serving as artificial synapses, as shown in Figure 4.

Desirable analog RS characteristics have been observed in a broad range of systems, including TiO_x, Ag/α-Si, WO_x, Ag_2S, and TaO_x based memristive devices [25–29]. Similarly, different types of biomimicking learning rules that are necessary for neuromorphic computing implementations have been demonstrated in memristor-based systems, depending on the internal dynamics of the cations or anions that drive the memristor's conductance change [1].

A typical implementation of the memristor-based neural network is the crossbar structure, shown in Figure 4, where every neuron on the left (layer 1) can be connected to every neuron at the bottom (layer 2) through plastic connections offered by the memristors in the columns associated with the output neurons.

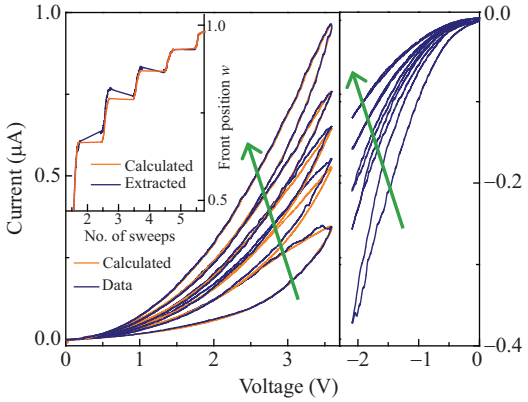

Figure 3 (Color online) Analog RS in a Ag/α-Si based memristive device, showing measured (blue) and calculated (orange) I-V characteristics of the device [27] @ Copyright 2010 American Chemical Society.

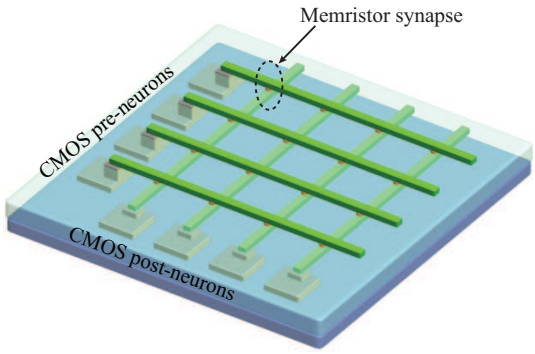

Figure 4 (Color online) Schematic of a memristor-based network structure. A memristor is formed at each cross-point and can simultaneously store data and process information, in ways similar to a biological synapse [27] @ Copyright 2010 American Chemical Society.

This approach offers high density and large connectivity (leading to high levels of parallelism) that are needed in neuromorphic hardware, and has been widely used by different research groups.

Biologically inspired learning rules such as spike-timing-dependent plasticity (STDP) and activity-dependent plasticity rules have already been demonstrated at the device levels [27, 30–32]. Figure 5(a) and (b) shows an example of a single memristor device acting as a synapse and reproducing the STDP behavior of biological synapses. Initial demonstrations of these rules were commonly aided by carefully engineered programming pulses applied to the devices [27], while recent studies have shown that the internal ionic dynamics (at both short-term and long-term time scales) in so-called second-order memristor devices [30, 31] and diffusive memristor devices [33] can naturally lead to different timing- and rate-based synaptic plasticity behaviors. Specifically in a second order memristor, the rise and decay of one state variable (e.g., local temperature) encodes the relative timing information of the voltage pulse inputs (spikes) and can subsequently modulate the change of a 2^{nd} state variable that represents the synaptic weight (e.g., filament size). These internal dynamic ionic effects allow the device to biofaithfully emulate both timing- and rate-dependent plasticity effects using simple, non-overlapping pulses found in biology [34, 35], and could further improve the system's computing efficiency beyond what's offered through the network topology. Other synaptic behaviors, including paired pulse facilitation, experience-dependent plasticity, frequency-dependent plasticity, heterosynaptic plasticity, and metaplasticity, have also been directly and naturally emulated with these dynamic memristive devices [31, 36, 37], and suggest memristor-based hardware systems can be extremely attractive in realizing bioinspired networks.

3 Memristor based neuromorphic computing systems

Neuromorphic hardware architectures typically comprise hybrid analog/digital circuits that implement physical models of neural computing systems, following computational principles analogous to the ones used by real nervous systems [7]. Prior efforts on neuromorphic computing systems implementation were based on mixed-signal CMOS technologies [38–42], acting as hardware accelerators for neuroscience model testing. These systems, however, are still based on separate memory and logic components and are to a large extent affected by the von Neumann memory bottleneck problem [43]. The ability to directly and physically map the network structure in memristor-based neural network approaches thus provides a welcome alternative. Below we discuss the operation principle and several recent demonstrations of memristor-based neuromorphic systems.

Mathematically, the operations of neuromorphic systems and many data-intensive tasks in general are based on vector-matrix multiplication processes, where the input vector is simultaneously compared with

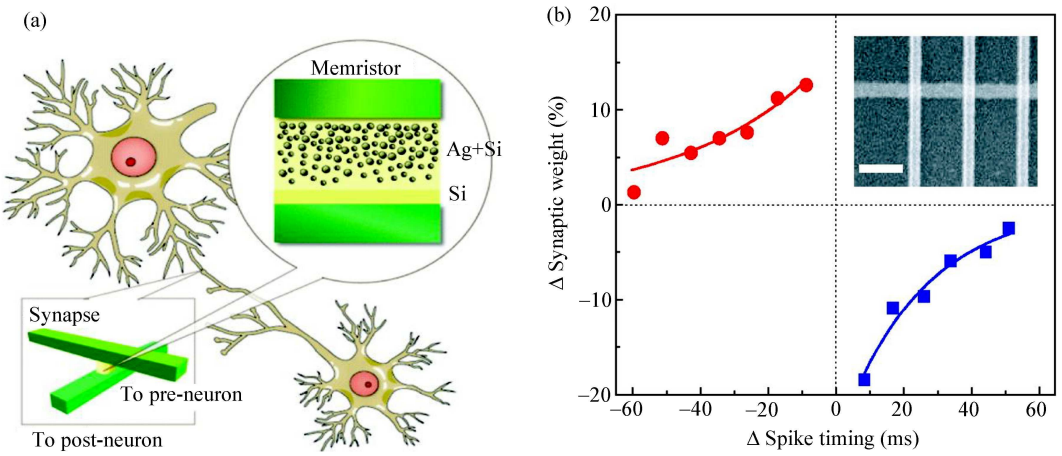

Figure 5 (Color online) (a) Mimicking the biological synapse using a single memristor device; (b) measured STDP behavior from a memristor [27] @ Copyright 2010 American Chemical Society.

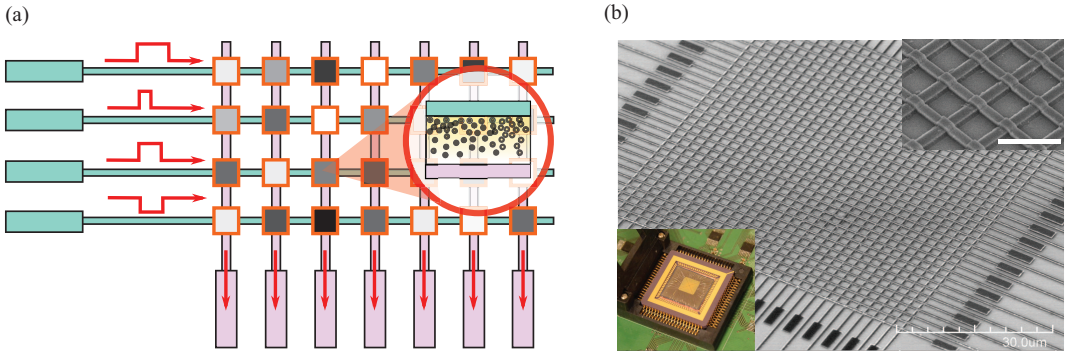

Figure 6 (Color online) (a) Schematic of a memristor crossbar-based computing system. The input neurons (green) and the leaky integrating output neurons (pink) are connected to the rows and columns of the crossbar, respectively. An output neuron collects currents flown from all memristor devices in the column associated with the neuron. The memristor devices can be programmed to different conductance states (represented in greyscale), through internal ion redistribution (inset). (b) Scanning electron micrograph (SEM) image of a fabricated memristor array and the memristor chip (lower left) [24] @ Copyright 2017 Springer Nature.

the stored vectors in the network through the vector-vector dot-product operation, and the neurons with better matches with the input will produce larger output values. The vector-matrix multiplication can be directly mapped on the memristor crossbar, as shown in Figure 6(a). Each memristor device at a cross-point represents a synapse linking an input neuron (at a given row) with an output neuron (at a given column), with the synaptic weight represented by the memristor conductance. If the input signal is coded by the amplitude (or width) of the applied input voltage pulse, the current (or charge) through the corresponding memristor device is then determined by the product of the input and the memristive synaptic weight [24]. As a result, the dot-product equation can now be naturally computed directly using Ohm's law (providing the multiplication function) and Kirchhoff's law (providing the summation function) in the device in physics. With this approach, the dot-product and vector-matrix multiplication operations are reduced to a single read operation in the memristor crossbar, represented by the output currents at the neurons that can be measured in parallel for all neurons in the network.

Several examples of memristor crossbar-based neuromorphic hardware have already been demonstrated in the last couple of years. They have been shown capable of performing different computing tasks including pattern classification, feature extraction, sparse coding, and recognition [11,44–48]. For example, the first memristor hardware performing pattern classification was demonstrated by a team at the University of California, Santa Barbara (UCSB), initially using a 2×10 crossbar [44] and later expanded to a 12×12 crossbar [11]. A generic dot-product engine using memristor arrays for neuromorphic applications was

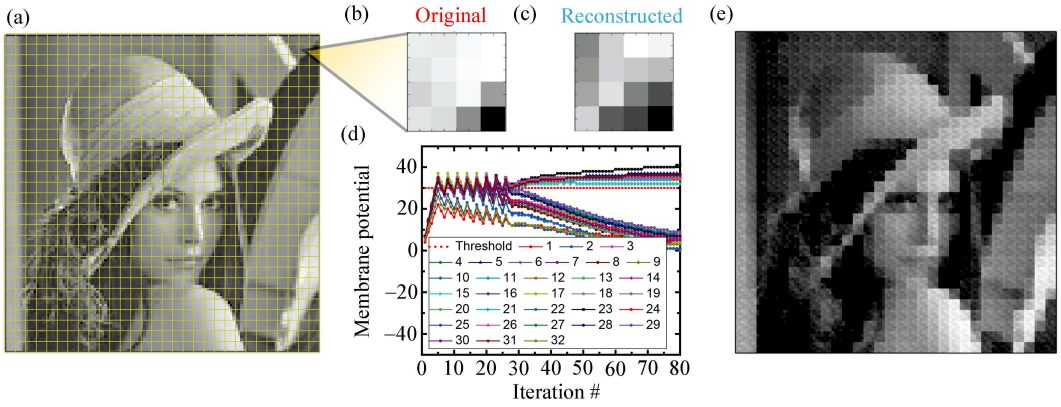

Figure 7 (Color online) Natural image processing using the memristor crossbar network. (a) The original 120×120 image to be processed is divided into non-overlapping 4×4 patches; (b) an example of a 4×4 patch from the original image; (c) the experimentally reconstructed patch from the 16×32 memristor crossbar; (d) membrane potentials of the neurons during sparse coding The red horizontal line marks the threshold parameter. Neurons having membrane potential above the threshold are active, while the other neurons will have exactly zero activity; (e) reconstructed image [24] @ Copyright 2017 Springer Nature.

introduced by Hewlett Packard Labs in 2016 [49]. In 2017, principal component analysis through online learning in a 2×8 crossbar [50] was demonstrated, along with a sparse coding chip that allows lateral neuron inhibition was developed at the University of Michigan using a 32×32 crossbar [24], shown in Figure 7. In the sparse-coding work, the memristor array is first used to learn features (receptive fields) from natural images. The learned features are then used to analyze the input image using a locally-competitive algorithm (LCA) to identify hidden features from the input and reconstruct the input with few active neurons based on the identified features.

4 Challenges and new directions

Several device-level challenges still need to be fully addressed before large-scale memristor-based neuromorphic hardware systems can be built. Fortunately, write/erase cycles, encountered during the network training phase, are typically infrequent during the lifetime of the network as most operations (e.g., vector-matrix multiplications) can be mapped into the read operation. On the other hand, most learning algorithms rely on the use of analog devices whose conductance needs to be updated in an incremental, instead of binary fashion [27]. The linearity and symmetry of the incremental conductance (synaptic weight) update have been shown to strongly impact the network performance and need to be optimized [45, 51]. One option to address this problem is through system level innovations, e.g., by using devices with fewer levels and assigning multiple devices to represent each weight. In this regard, binary memristive devices can possibly be adopted in neural network implementations, but at the expense of the effective weight density.

An attractive property of artificial neural network applications is their ability to tolerate device level imperfections. For instance, device variability may not be a significant problem for a memristor based neural network, since the adaptive evolution nature of the network allows the device variations to be factored in during learning [52,53]. In a recent study focusing on image analysis based on the sparse coding algorithm, it was found that moderate device-to-device variations may even help the image reconstruction performance by introducing high spatial frequency features in the dictionary, particularly at small network sizes [53].

Besides acting as synaptic elements, recent studies have also shown that memristors can provide neuronal functions that can collect (integrate) input signals and generate stochastic spikes based on the integrated inputs [54, 55]. Networks of dynamic memristors can also exhibit emergent behaviors that are not expected from individual devices, due to the highly non-linear interactions of the devices. As

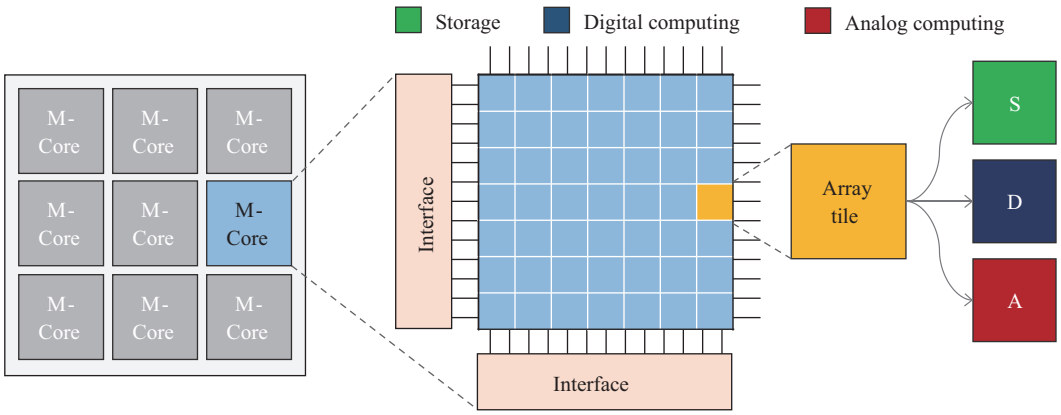

Figure 8 (Color online) Block diagram showing the different layers that can be reconfigured in a possible reconfigurable in-memory computing architecture [5] @ Copyright 2017 IEEE.

the internal dynamics of the devices can be accurately modeled [30, 31] while the interactions may be measured and reliably simulated through circuit parameters, emergence of functionality in these dynamic networks may be systematically predicted and analyzed. This type of analysis can thus lead to new algorithm developments that can efficiently take advantage of the network dynamics.

Memristor devices have also been used to efficiently perform functions such as arithmetic/analog processing [56]. Combined, these three functions, memory/storage, arithmetic and neuromorphic/artificial networks, cover a broad range of tasks that are encountered today from small systems such as sensor nodes to large ones such as server farms. We note that these three functions can in fact be realized using a common physical fabric based on memristor crossbar arrays [5], allowing the development of a general computing system whose functions and resources can be dynamically reconfigured based on the task and data patterns.

One such approach is to build around the concept of reconfigurable memristor crossbar-based cores [5], as shown in Figure 8. Each core is identical and can be further freely divided into smaller modules down to the finest grain – device level. Both the cores and the modules can be (re)configured through software to perform one of the three system tasks discussed above, as shown in Figure 8. Such an in-memory computing system can offer three desirable properties: first, the ability to process an arbitrary workload in its optimal computing domain (arithmetic, analog or neural networks); second, a high degree of scalability and reconfigurability (both coarse-grain and fine-grain) enabled by the common physical fabric and the modular design; and finally, true in-memory computing at the lowest physical level with minimal data migration.

The challenges in physically implementing such a general in-memory computing architecture include identifying and fabricating memristor devices and arrays that support the different modes of operations. For instance, binary devices are more suitable for storage and digital vector operations, while multilevel devices are preferred in neuromorphic operations and can provide higher effective storage density. In any case, devices with high ON/OFF and low programming current will be preferred to improve the system's performance and power efficiency.

It may be argued that such a memristor-based in-memory computing system follows the natural evaluation of computing systems from central processing units (CPUs) to graphics processing units (GPUs). Typically, CPUs rely on large (coarse-grain) processing cores to provide the computational power but suffer significantly from the memory bottleneck. On the other hand, GPUs rely on finer-grain cores to improve parallelism with faster local memory access. By merging memory and processing units together at the finest grain level – device level, the memristor-based system can fundamentally eliminate the memory bottleneck and offer high parallelism and dynamic reconfigurability, making it a viable solution for future computing needs.

5 Conclusion

The last a few years have witnessed great process in the development of memristive devices and computing systems that can efficiently exploit the properties of such devices. Memristive-based hardware systems have shown excellent potential to meet the computing needs past the von Neumann/Moores-Law era. Carefully designed memristor devices can natively mimic the dynamics of their biological counterpart - synapses, and allow the network to develop emergent behaviors not expected from single device properties. Memristor crossbars provide a natural platform to implement massively-parallel and power efficient vector-matrix operations that form the basis of neuromorphic operations and other data-intensive tasks. It is expected that advances in device technology and architecture developments can eventually lead to large-scale hardware implementation of neuromorphic computing systems. However, to turn these ideas into reality will require dedicated, multidisciplinary research branching materials, devices, architecture and algorithm developments.

Acknowledgements This work was supported in part by National Science Foundation (NSF) (Grant Nos. ECCS-1708700, CCF-1617315). We would like to thank F CAI, J LEE and J SHIN for helpful discussions.

References

1 Yang J J, Strukov D B, Stewart D R. Memristive devices for computing. Nat Nanotech, 2013, 8: 13–24
2 Kim K H, Gaba S, Wheeler D, et al. A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. Nano Lett, 2012, 12: 389–395
3 Pershin Y V, Di Ventra M. Neuromorphic, digital, and quantum computation with memory circuit elements. Proc IEEE, 2012, 100: 2071–2080
4 Gaba S, Knag P, Zhang Z Y, et al. Memristive devices for stochastic computing. In: Proceedings of IEEE International Symposium on Circuits and Systems, Melbourne, 2014. 2592–2595
5 Zidan M, Jeong Y J, Shin J H, et al. Field-programmable crossbar array (FPCA) for reconfigurable computing. IEEE Trans Multi-Scale Comput Syst, 2017. doi: 10.1109/TMSCS.2017.2721160
6 Borghetti J, Snider G S, Kuekes P J, et al. 'Memristive' switches enable 'stateful' logic operations via material implication. Nature, 2010, 464: 873–876
7 Mead C. Neuromorphic electronic systems. Proc IEEE, 1990, 78: 1629–1636
8 Indiveri G, Horiuchi T K. Frontiers in neuromorphic engineering. Front Neurosci, 2011, 5: 118
9 Chicca E, Stefanini F, Bartolozzi C, et al. Neuromorphic electronic circuits for building autonomous cognitive systems. Proc IEEE, 2014, 102: 1367–1388
10 Gaba S, Sheridan P, Zhou J, et al. Stochastic memristive devices for computing and neuromorphic applications. Nanoscale, 2013, 5: 5872–5878
11 Prezioso M, Merrikh-Bayat F, Hoskins B D, et al. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. Nature, 2015, 521: 61–64
12 Indiveri G, Linares-Barranco B, Legenstein R, et al. Integration of nanoscale memristor synapses in neuromorphic computing architectures. Nanotechnology, 2013, 24: 384010
13 Zidan M A, Chen A, Indiveri G, et al. Memristive computing devices and applications. J Electroceram, 2017, 39: 4–20
14 Chua L O, Kang S M. Memristive devices and systems. Proc IEEE, 1976, 64: 209–223
15 Strukov D B, Snider G S, Stewart D R, et al. The missing memristor found. Nature, 2008, 453: 80–83
16 Govoreanu B, Kar G S, Chen Y Y, et al. 10×10 nm^2 Hf/HfO_x crossbar resistive RAM with excellent performance, reliability and low-energy operation. In: Proceedings of IEEE International Electron Devices Meeting, Washington, 2011
17 Torrezan A C, Strachan J P, Medeiros-Ribeiro G, et al. Sub-nanosecond switching of a tantalum oxide memristor. Nanotechnology, 2011, 22: 485203
18 Lee M J, Lee C B, Lee D, et al. A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structures. Nat Mater, 2011, 10: 625–630
19 Valov I, Lu W D. Nanoscale electrochemistry using dielectric thin films as solid electrolytes. Nanoscale, 2016, 8: 13828–13837
20 Younis A, Chu D, Lin X, et al. High-performance nanocomposite based memristor with controlled quantum dots as charge traps. ACS Appl Mater Interface, 2013, 5: 2249–2254
21 Stoliar P, Rozenberg M, Janod E, et al. Nonthermal and purely electronic resistive switching in a Mott memory. Phys Rev B, 2014, 90: 45146
22 Wong H S P, Raoux S, Kim S B, et al. Phase change memory. Proc IEEE, 2010, 98: 2201–2227
23 Diao Z T, Li Z J, Wang S Y, et al. Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory. J Phys-Condens Matter, 2007, 19: 165209
24 Sheridan P M, Cai F X, Du C, et al. Sparse coding with memristor networks. Nat Nanotech, 2017, 12: 784–789

25 Chang T, Jo S H, Kim K H, et al. Synaptic behaviors and modeling of a metal oxide memristive device. Appl Phys A, 2011, 102: 857–863
26 Hasegawa T, Ohno T, Terabe K, et al. Learning abilities achieved by a single solid-state atomic switch. Adv Mater, 2010, 22: 1831–1834
27 Jo S H, Chang T, Ebong I, et al. Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett, 2010, 10: 1297–1301
28 Kim S, Choi S H, Lu W. Comprehensive physical model of dynamic resistive switching in an oxide memristor. ACS Nano, 2014, 8: 2369–2376
29 Seo K, Kim I, Jung S, et al. Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device. Nanotechnology, 2011, 22: 254023
30 Kim S, Du C, Sheridan P, et al. Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity. Nano Lett, 2015, 15: 2203–2211
31 Du C, Ma W, Chang T, et al. Biorealistic implementation of synaptic functions with oxide memristors through internal ionic dynamics. Adv Funct Mater, 2015, 25: 4290–4299
32 Kuzum D, Yu S, Wong H S. Synaptic electronics: materials, devices and applications. Nanotechnology, 2013, 24: 382001
33 Wang Z R, Joshi S, Savelev S E, et al. Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. Nat Mater, 2017, 16: 101–108
34 Zidan M A, Jeong Y J, Lu W D. Temporal learning using second-order memristors. IEEE Trans Nanotechnol, 2017, 16: 721–723
35 Ma W, Chen L, Du C, et al. Temporal information encoding in dynamic memristive devices. Appl Phys Lett, 2015, 107: 193101
36 Zhu X, Du C, Jeong Y J, et al. Emulation of synaptic metaplasticity in memristors. Nanoscale, 2017, 9: 45–51
37 Yang Y, Chen B, Lu W D. Memristive physically evolving networks enabling the emulation of heterosynaptic plasticity. Adv Mater, 2015, 27: 7720–7727
38 Merolla P A, Arthur J V, Alvarez-Icaza R, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science, 2014, 345: 668–673
39 Benjamin B V, Gao P, McQuinn E, et al. Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. Proc IEEE, 2014, 102: 699–716
40 Furber S B, Galluppi F, Temple S, et al. The SpiNNaker project. Proc IEEE, 2014, 102: 652–665
41 Schemmel J, Briiderle D, Griibl A, et al. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In: Proceedings of IEEE International Symposium on Circuits and Systems, Paris, 2010. 1947–1950
42 Pfeil T, Grübl A, Jeltsch S, et al. Six networks on a universal neuromorphic computing substrate. Front Neurosci, 2013, 7: 11
43 Indiveri G, Liu S C. Memory and information processing in neuromorphic systems. Proc IEEE, 2015, 103: 1379–1397
44 Alibart F, Zamanidoost E, Strukov D B. Pattern classification by memristive crossbar circuits using ex situ and in situ training. Nat Commun, 2013, 4: 2072
45 Sheridan P M, Du C, Lu W D. Feature extraction using memristor networks. IEEE Trans Neural Netw Learning Syst, 2016, 27: 2327–2336
46 Choi S, Sheridan P, Lu W D. Data clustering using memristor networks. Sci Rep, 2015, 5: 10492
47 Sheridan P, Ma W, Lu W. Pattern recognition with memristor networks. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, 2014. 1078–1081
48 Adhikari S P, Yang C J, Kim H, et al. Memristor bridge synapse-based neural network and its learning. IEEE Trans Neural Netw Learning Syst, 2012, 23: 1426–1435
49 Hu M, Strachan J P, Grafals E M, et al. Dot-product engine for neuromorphic computing. In: Proceedings of the 53rd Annual Design Automation Conference, Austin, 2016
50 Choi S, Shin J H, Lee J, et al. Experimental demonstration of feature extraction and dimensionality reduction using memristor networks. Nano Lett, 2017, 17: 3113–3118
51 Yu S, Chen P Y, Cao Y, et al. Scaling-up resistive synaptic arrays for neuro-inspired architecture: challenges and prospect. In: Proceedings of International Electron Devices Meeting, Washington, 2015
52 Sheridan P, Lu W D. Defect consideratons for robust sparse coding using memristor arrays. In: Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures, Boston, 2015, 137–138
53 Ma W, Cai F, Du C, et al. Device nonideality effects on image reconstruction using memristor arrays. In: Proceedings of 2016 IEEE International Electron Devices Meeting (IEDM), San Francisco, 2016
54 Kumar S, Strachan J P, Williams R S. Chaotic dynamics in nanoscale NbO_2 Mott memristors for analogue computing. Nature, 2017, 548: 318–321
55 Tuma T, Pantazi A, Le Gallo M, et al. Stochastic phase-change neurons. Nat Nanotech, 2016, 11: 693–699
56 Chen B, Cai F X, Zhou J T, et al. Efficient in-memory computing architecture based on crossbar arrays. In: Proceedings of International Electron Devices Meeting, Washington, 2015