

RRAM-Based Hardware Implementations of Artificial Neural Networks: Progress Update and Challenges Ahead

M. Prezioso, F. Merrih-Bayat, B. Chakrabarti, and D. Strukov

Department of Electrical and Computer Engineering, University of California at Santa Barbara
Santa Barbara, California 93106, USA

ABSTRACT

Artificial neural networks have been receiving increasing attention due to their superior performance in many information processing tasks. Typically, scaling up the size of the network results in better performance and richer functionality. However, large neural networks are challenging to implement in software and customized hardware are generally required for their practical implementations.

In this work, we will discuss our group's recent efforts on the development of such custom hardware circuits, based on hybrid CMOS/memristor circuits, in particular of CMOL variety. We will start by reviewing the basics of memristive devices and of CMOL circuits. We will then discuss our recent progress towards demonstration of hybrid circuits, focusing on the experimental and theoretical results for artificial neural networks based on crossbar-integrated metal oxide memristors. We will conclude presentation with the discussion of the remaining challenges and the most pressing research needs.

Keywords: neuromorphic computation, neural networks, memristive devices, RRAM

1. INTRODUCTION

Artificial Neural Networks (ANNs) date back to the 40's starting with the work of McCulloch and Pitts[1] and surviving vicissitudes they finally found their deserved place as one of the most promising computing paradigm beyond Von Neumann architecture[2]. This is due not only to the advancement of the theoretical tools giving birth to many interesting applications ranging from data clustering to image classification[3, 4], drug activity prediction[5], and analysis of particle accelerator data[6]; but also to the availability of computational power suitable for handling the massive amount of computations required for software implementations of practical artificial neural networks. However, this computational capability comes at a price of huge power consumption, which is due to the inadequacy of Von-Neumann architectures for running the biologically inspired ANN algorithms. A classic example is that our brain consumes about ten Watts at peak activity, whereas a computer would need MWs of power in order to reproduce even few seconds of a rat brain activity[7]. Although not being quite a fair comparison, this example does point to the scale of the problem. It appears clear, therefore, that there is a need for a change in the hardware architecture in order to achieve new level of efficiency for the aforementioned applications.

Indeed, the common major challenge in building ANNs is efficient vector-by-matrix multiplication, which in turn requires compact implementation of synaptic weights. The purely CMOS based circuits are insufficient for meeting these challenges, mostly because CMOS-implemented synapses are way too bulky. One promising solution is to process information in analog form using hybrid CMOL (**CMOS + Molecular-scale devices**) circuits combining the existing CMOS integrated circuits with crossbar add-ons using resistive switching ("memristive") devices [8]. Recent experiments and theoretical estimates based on realistic simulations have shown that such hybrid circuits, being scaled down to realistic 15-nm dimensions, may provide a 1,000× increase in speed and energy-efficiency in comparison with similar networks implemented using conventional circuits.

We will next briefly introduce the memristive devices and CMOL basic concept and then review our recent experimental results.

2. EXPERIMENTAL RESULTS

2.1 Memristive Devices and Crossbar Circuits

The distinctive feature of memristive devices is to possess some kind of memory. The term ‘Memristive’ was coined by Leon Chua in the 70’s[9], following his work on non-linear circuits. In the vast set of memory devices, we focus here on Resistive RAMs (RRAMs), and in particular on Valence Change Memory (VCM) devices[10]. Such systems are composed of two-terminal devices in which a switching layer is sandwiched between two electrodes in a metal-insulator-metal structure. In this type of memory devices, the switching layer is normally a transition metal oxide, and the switching mechanism is based on ion migration which involve drift of oxygen ions under the influence of an applied electric field. The ionic transport is temperature activated and the ion diffusion is greatly enhanced by the Joule heating due to the electron current passing through the film[11]. For example, the inset of figure 1a shows the material stack of the devices which was used in our work. Platinum and titanium are used for the bottom and top electrodes, respectively. The 30-nm TiO_{2-x} oxide is the switching layer, while the thin 4-nm alumina layer is a barrier, which was optimized to increase the device I-V non-linearity.

Normally, a ‘forming’ process (dielectric breakdown) is necessary in order to enable the switching operation. During this process, one or more switching filaments are formed. The filaments consist of regions with defects which promote the drift of oxygen ions. Oxygen vacancies are created during the forming process, resulting in the change in cation valence, effectively doping the material and changing its conductivity. Figure 1a shows the I-V curve of the forming process (green line), with the forming occurring at 2 V and 180 μA . Typically, the forming process requires much higher electrical stress and in order to reduce the forming voltage in our devices the stoichiometry of the TiO_{2-x} had to be optimized. Once the forming has occurred, the device jumps to the low resistance state (LRS) and, from now on, it can be programmed. In the same figure, the red curve shows the ‘Reset’ cycle, during which the device is turned to its high resistance state (HRS). The threshold for the ‘Reset’ is ~ -1 V, and this switching transition is not abrupt. The fabricated devices are effectively analog non-volatile memory because their resistive state can be changed continuously. Applying ‘Reset’ voltages below -1 V but higher than -2 V will turn the device into an intermediate resistive state, while an applied voltage below -2.0 V turns the device completely off. This analog behavior is showed in figure 1b. The graph and the zoomed inset, show the different resistance states to which the device was effectively tuned (y-axis) vs. the desired target resistance (x-axis). The number of states is 32 over slightly more than 1 decade in resistance. Finally, the blue curve on Figure 1a shows the ‘Set’ cycle turning the device from HRS to LRS. It shows that the ‘Set’ process is much more abrupt than the Reset.

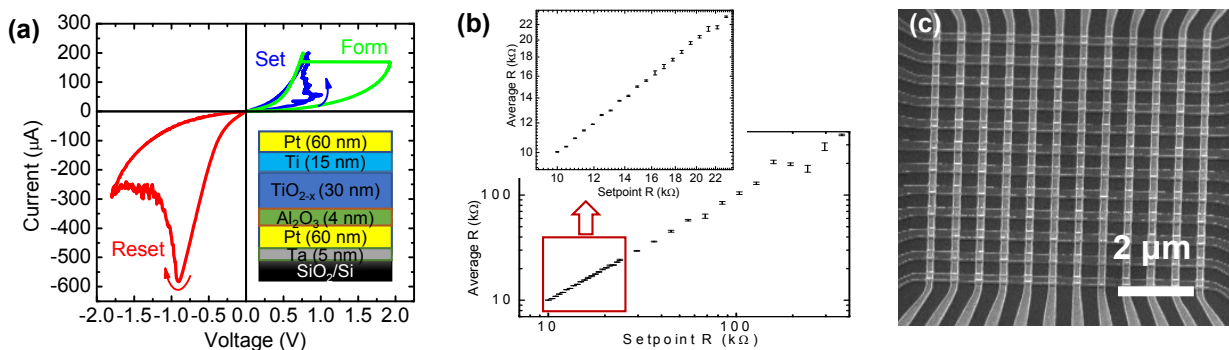


Figure 1. $\text{Al}_2\text{O}_3/\text{TiO}_{2-x}$ resistive switching devices: (a) typical quasi-dc I-V curves of memristor forming and switching, with the inset showing the device stack. (b) Analog properties of crossbar-integrated devices: tuning of the resistance measured at a non-disturbing voltage of 0.1 V (sampled at 1 kHz and averaged over 200 s interval) to various values within the dynamic range. (c) Micrograph of a 12×12 -crosspoint crossbar implemented with DUV lithography.

Figure 1c shows the devices integrated in a passive 12 rows (bottom electrodes) by 12 columns (top) crossbar structure. The line width is 200 nm and the distance between the lines is 400 nm. In such a passive crossbar, there are no dedicated active devices (e.g. transistors) implementing selector functionality which is required to enable

unique access of peripheral circuitry to a specific crosspoint device. The advantage of such structure is the low fabrication complexity and small area, but there are also several challenges. One challenge is that memristors must have low forming voltages to ensure successful forming of all devices in the crossbar circuits. The other two important issues are the so-called sneak path and half-select disturbance[12]. The former problem is due to current leaking through devices connected in parallel during the read, programming and forming operations. The second problem is related to the disturbance of the state of the devices that share one of the electrodes with the device being programmed. To minimize the effects of sneak paths and half-select disturbances, three different biasing schemes were employed (figure 2).

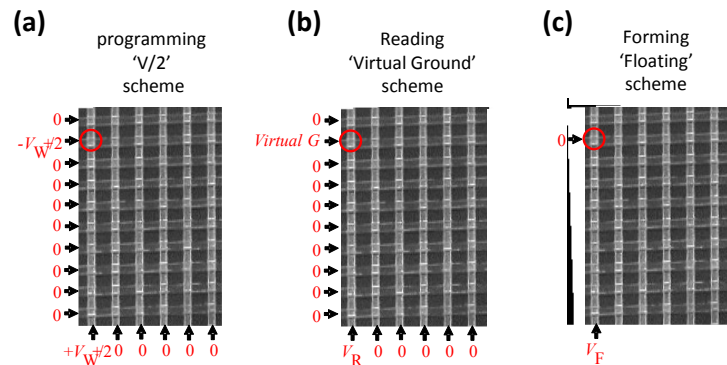


Figure 2. Biasing schemes used during various mode of operations. (a) Programming: a ' $V/2$ ' scheme is used to guarantee that no voltage higher than $|V/2|$ drops over any device different from the selected one. This is needed in order to reduce the perturbation of unselected or half-selected devices. (b) Reading: in this case, virtual ground is used on the selected row (a source measure unit is used to generate 0 V and measure current), the reading voltage (V_R) is applied to selected column and all the other terminals are grounded. In this way, only the current coming from the selected device will be read. (c) Forming: here a simple 'floating' scheme is adopted. Assuming that after each forming event the formed device is Reset to HRS, which is slightly lower than pre-forming resistance, the maximum voltage drop during forming of the device on its nearest neighbors will be between $V/3$ and $V/2$. At the same time, the total current will be less as compared to that of the two previous schemes. In all panels, the red circle indicates the selected device.

A great deal of work has been already done in engineering the memristive devices, integrating them into crossbars, and investigating possible applications of such circuits. However, in order to be able to exploit the advantages of crossbar-integrated memristive devices, a good strategy for their integration with CMOS technology is also necessary. CMOL hybrid circuits were conceived to address this problem and, more specifically, to provide efficient interface between crossbars and CMOS layer[13]. The basic idea is to stack one or more layers[14] of memristive crossbars on top of the CMOS circuitry (figure 3). CMOL's key feature is area-distributed interface, which is implemented with tilted and segmented crossbar and double decoding scheme. Such interface allows to uniquely address every crosspoint device without the need for overlay alignment between the nano and the CMOS subsystems, which can potentially have dramatically different critical dimensions.

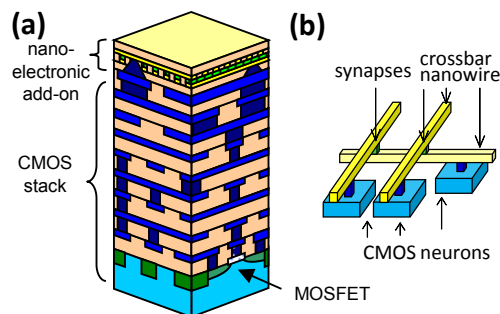


Figure 3. CMOL circuits. (a) A cartoon of a hybrid CMOS/memristor integrated circuit; (b) the example of three CMOS cells (neurons) interconnected via corresponding crossbar nanowires (dendrites and axons) and crosspoint memristive devices (synapses), which are located above CMOS layer.

Various CMOL circuit architectures have been investigated in the context of digital memory and logic circuits[15]. However, the main appeal of CMOL seems to be in neuromorphic applications. To implement ANNs, CMOS circuits are typically arranged as an array of cells with each implementing a neuron functionality. The CMOL topology would then naturally enable large connectivity between CMOS-implemented neurons via crossbar wires and crosspoint memristive devices, playing the role of artificial synapses (figure 3b). Thus CMOL circuit with nanoscale memristive device could address some of the major implementation challenges of neural networks, such as practical integration of large number of synapses and ensuring large connectivity between neurons. Moreover, as we will show, CMOL circuits would naturally implement vector-by-matrix multiplication, which is the main bottleneck operation of ANNs.

2.2 Fire-Rate Neuromorphic Networks: Single Layer Perceptron

To demonstrate potentials of memristive devices in neuromorphic computing, we have experimentally demonstrated, for the first time, artificial neural network based on a passive integrated memristive crossbar[16] (figure 4). In the experiment memristors were used as synapses in a Single Layer Perceptron (SLP) classifier[17], which is essentially a single-layer feed-forward network. The network inputs were connected through the synapses to the output neurons, and each output neuron computed a weighted sum of the network inputs as follows:

$$T_i = \sum_k w_{ik} I_k \quad (1)$$

Here T_i is the total input of neuron i and w_{ik} is the weight of a synapse connecting the k -th input and i -th output neuron. Note that in case of $i > 1$ the equation 1 describes a vector-by-matrix multiplication between input vector and matrix of weights. The outputs of the neurons, which are also the outputs of the network in the case of SLP, are obtained by applying nonlinear “activation” function, such as hyperbolic tangent, to the weighted sum

$$O_i = f(T_i) \quad (2)$$

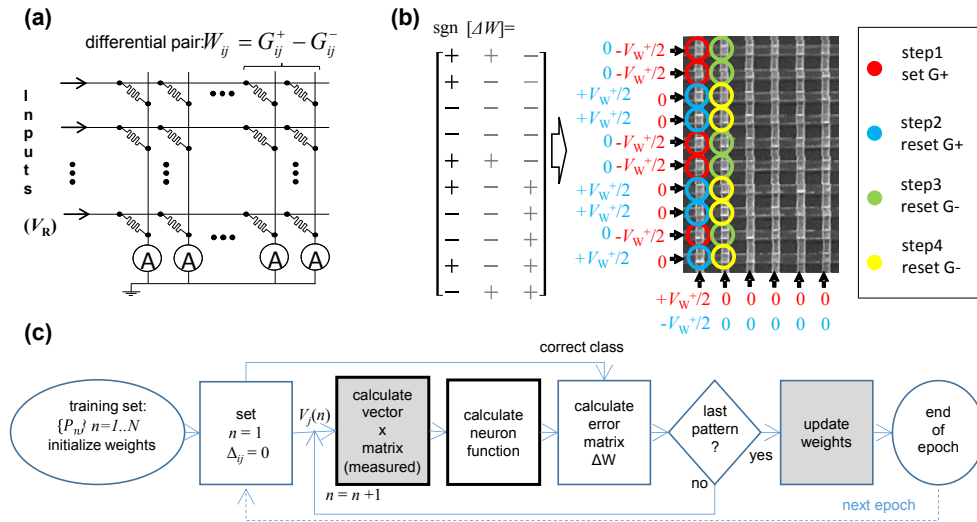


Figure 4. (a) An implementation of vector-by-matrix multiplication using a memristive crossbar. Here we use a pair of weights to implement one synapse to enable negative weight values. (b) The first four steps of crossbar update. The sign of the weight increment matrix (on the left), which is first obtained after one epoch of training, specifies the direction of the state update for each device in crossbar circuit, i.e. whether to incrementally set or reset the device. The update is performed using the V/2 scheme with appropriate chosen voltage (on the right). The voltage shown in red and blue are for the first and second step, respectively. (c) Flow chart of the training algorithm. Grey boxes show the steps implemented in hardware while all remaining steps were emulated in software. The outlined boxes show the only steps needed for operation of the neural network, i.e. after it has been successfully trained.

Naturally, the output O_i of the neuron i would be higher than a certain threshold (normally taken as zero) only for a subset of the possible input vectors. The key point is that if the weight vector is correctly selected, the neuron will be able to identify the desired subset of input vectors, or, in other words, to classify them. The process of finding the correct weight is called training, and the algorithm used in the process is the *training algorithm*. In the so-called supervised training, the algorithm is based on comparing the actual neuron output with the desired output in order to obtain the error and to minimize it by adjusting the weights [17] (figure 4c).

We used supervised training to teach the experimental metal oxide memristor-based neural network to perform the perfect classification of 3×3 -pixel black/white images into 3 classes, one per output neuron. In particular, the patterns were the stylized versions of Z, V and N characters on the 3×3 grid. We used both the ideal patterns and the noisy ones created by flipping one pixel in the image. This resulted in 10 patterns per class for a total of 30 patterns (figure 5a). In the classifier's operation, the vector-by-matrix multiplication of the input signals by weights is performed on the physical level (in the analog domain) as a result of using Ohm's and Kirchhoff's laws. In particular, the inputs are represented with voltages and the weights with device conductances. By applying the input voltages to crossbar's row lines and reading out the currents flowing into virtually grounded column lines, the vector-by-matrix is computed as shown in figure 4a. More precisely, input voltages will be $+V_R$ or $-V_R$, depending on the color of the pixel, and the biasing scheme will be the 'virtual ground' one (figure 2b). Note that, in order to represent negative weights, the synapse is implemented by a pair of devices; the current of the column containing the negative weights is subtracted from the current coming from the column having the positive weights.

The required area in the crossbar was 10×6 devices: 10 rows were sufficient for processing 3×3 pixels and applying bias input, which is required for the convergence of the training algorithm. Six columns were used to collect currents for three neurons. During the training phase, all the patterns were fed into the network and the output was used to calculate the error and update the state of devices using a coarse-grain variety of the Delta Rule training algorithm[17]. The flow chart of the algorithm is shown in figure 4c. It is essential to note that this algorithm does not need to know the actual conductance values of the devices in order to converge. After the total error matrix has been determined, weight updates were performed at one column of devices at a time using 'V/2 scheme', resulting in four update steps for two columns (figure 4b). On average, the network reached the perfect classification on the training set after ~ 20 epochs (one epoch is one complete crossbar update), as shown in figure 5b. For further details please see ref.[16].

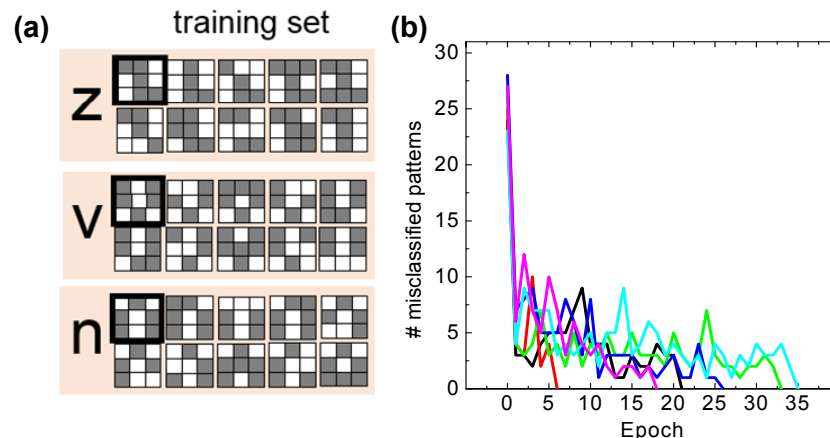


Figure 5. Pattern classifier. (a) Training-set patterns. The three classes are represented by the stylized characters X, V and Z plus their noisy versions obtained by flipping one pixel in the 3×3 images. (b) Experimental results for six different runs of the training for the SLP. The y-axis represents the number of misclassified patterns (out of 30 total) at each epoch. In all cases perfect classification was reached.

Note that the demonstrated SLP is particularly significant because it represents the basic element of the much more complex and powerful deep learning convolutional neural network, which are now widely used in many practical applications.

2.3 Spike-Timing Dependent Plasticity

In the previously considered firing-rate artificial neural networks, neuron spiking activity was represented only by its average rate, which was encoded as a voltage. In more advanced spiking neural networks (SNN) spiking activity is modeled explicitly. The general motivation for SNN is attributed to much higher information content which can be carried by explicit spiking signals[18, 19], much better energy efficiency[20, 21] and potentials for coordinated processing of spatial and temporal information[22, 23]. Additional motivation comes from the biology, which uses spiking signals and Spike-Timing Dependent Plasticity (STDP) to implement Hebbian learning[24, 25]. In this algorithm, when a pre-synaptic neuron A (see figure 6a) fires (spikes) before the post-synaptic neuron B, it means that a causal relation between the two exists and the synapse connecting those neurons is reinforced. In the other case, the synapse is depressed, as shown in the image in figure 6b. The latter represents what is called an ‘STDP window’, expressing the dependence of the synaptic weight update on the difference of arrival times of spikes. The one showed in the figure 6b was the first discovered in biology in ref. [24] and is widely accepted as the main learning mechanism in the brain (although many other STDP windows are present in the brain’s cortex [26]).

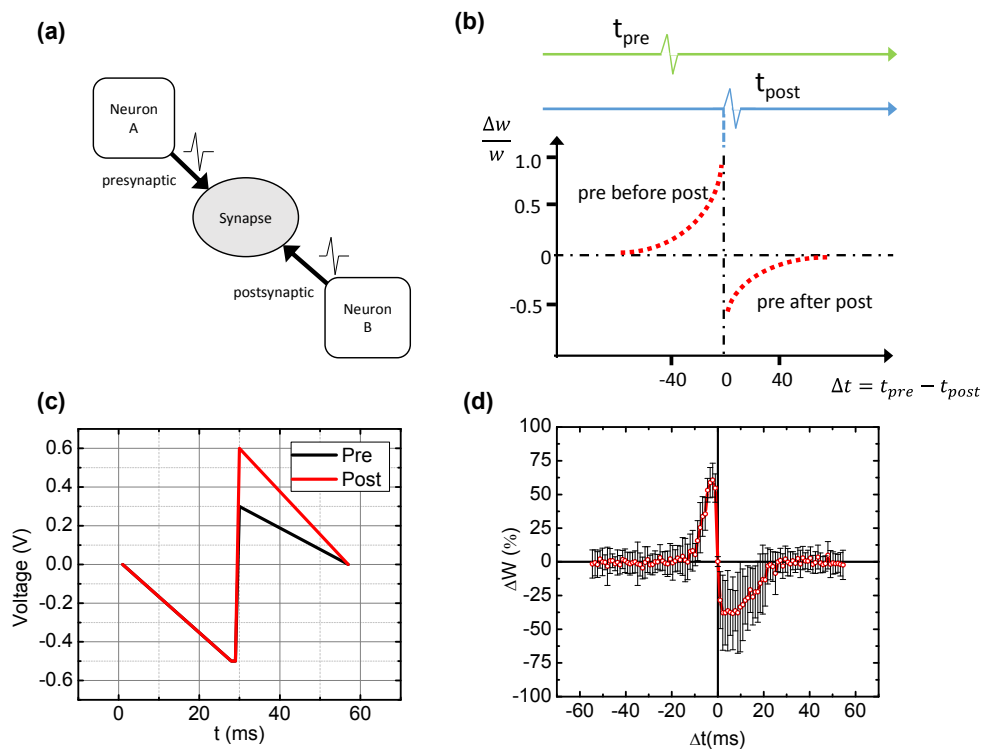


Figure 6. Spike-Timing Dependent Plasticity. (a) Definition of pre- and post-synaptic neurons. (b) The STDP window (shown schematically) similar to the one originally found in ref.[24]. (c) Amplitude and duration of pre (black) and postsynaptic (red) experimental pulses. (d) Experimental results which reproduced the biological STDP window shown in panel (b).

Recently, we have demonstrated that it is possible to reproduce such STDP window with a single TiO_{2-x} memristive device used as an artificial synapse. In particular, using the ‘V/2 scheme’ in figure 2a, we applied the specific presynaptic pulses to the row and the postsynaptic one to the column of the selected device. Amplitudes and shapes of the pulse are shown in figure 6c. First the device was tuned to an initial state close to 30 kOhm, roughly in the middle of its dynamical range. Then, the pulses were applied with different delays, from -60 ms to 60 ms and the

changes in resistance of the device was recorded. In particular, every data point on figure 6d, shows the normalized change in the resistance with respect to the initial state of the device, averaged across 10 runs, for a particular delay between pre- and post-synaptic pulses. The initial state was kept constant, i.e. if the resistance changed by more than 10%, we programed the device back to 30 kOhm – see refs.[27, 28] for more details. Setting the device back to the initial state was necessary because the change in the device resistance is a function of the resistance itself. Such dependence of the synaptic weight change on the actual weight has been found in biology as well, and has profound consequences on the dynamics of the spiking neural networks[29-31].

3. FUTURE STEPS AND CHALLENGES

The next step resides in the monolithic integration of memristive crossbars on top of the CMOS layer (see figure 7a). The critical factors for successful monolithic integration are (i) to create functional electrical interconnection between the memristive crossbar and the CMOS and (ii) to have the crossbar and the CMOS fabrication processes to be compatible to each other. Moreover, the memristive components need to be electrically compatible with the CMOS circuitry in terms of operational voltage and current requirements. Memristive components have been previously integrated between two metal layers of a CMOS process[32, 33]. However, vertical monolithic integration requires flexibility of the process so that memristive crossbars can be integrated on any pre-fabricated CMOS chip. This is much more challenging as the two fabrication processes would be completely independent of each other and successful demonstration of fully functional crossbars monolithically integrated on CMOS chip is yet to be reported. We are currently working on the development of a vertical monolithic integration process that can achieve reliable performances from the integrated devices. The main issues are to optimize device characteristics in order to achieve electrical compatibility (voltage and current requirements) with the CMOS as well as creating successful electrical connection with the underlying CMOS. In particular, it would be beneficial to find the right planarization process in order to overcome the severe topography on the surface of a CMOS chip (figure 7b-c).

Moreover, many other more fundamental problems and issues remain to be solved, even for the realization of simple networks. Starting from a device level, the switching mechanism on which RRAMs are based has an intrinsic high level of stochasticity. As a consequence, a wide scattering in the switching dynamics (thresholds, speed), even across a single crossbar, is present. This can give rise to issues during the learning phase, in the form of a very different updates (learning rate) among devices, or, worse, can cause some devices to be stuck in ON or OFF states[34]. The good news is that artificial neural networks are inherently resilient to failures in some of the synapses, so this problem can be mitigated by networks topology and redundancy.

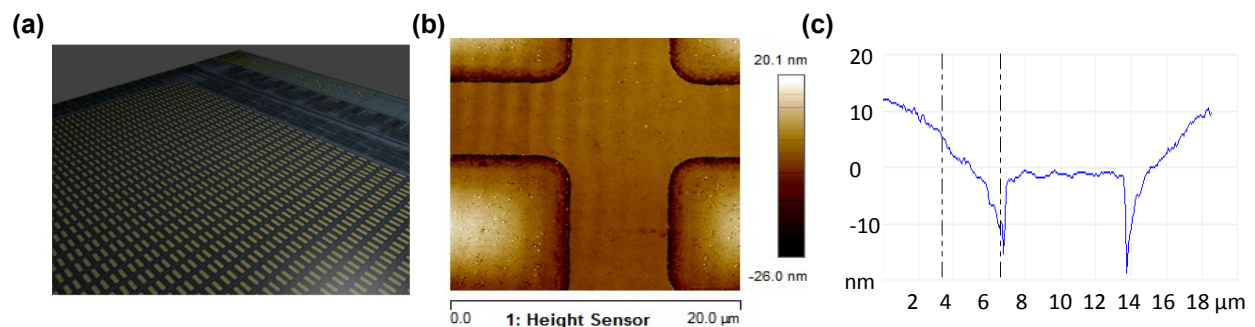


Figure 7. CMOS integration. (a) 3D image showing the pads on top of the CMOS layer of the chip designed to test the integration. (b) AFM image of the region between the pads. (c). AFM profile taken from the image in panel (b).

Another problem stems from the lack of active selectors in the crossbars. Current leakage is the main drawback of passive crossbars, both for the possible perturbation of the resistance state of unselected devices, and in terms of energy efficiency. In order to reduce those problems, engineering higher non-linearity in device I-V curve presents one possible solution. In the presented work, we introduced such non-linearity by means of the alumina barrier between the bottom electrode and the switching layer. While it was sufficient for our experiment, higher non-

linearity would be desirable for more practical larger crossbar circuits. Passive selector for crossbar circuits is still an open field of research, and few example can be found in refs.[35-38].

Lastly, due to the lack of extensive experimental work, there is still much to be learnt about training and operation of ANNs based on memristive devices. The intrinsic non-linearity and stochasticity of such devices complicate the already complex behavior of the ANNs as dynamical systems. Simulation work give some insights to these issues but experimental validations are still lacking.

4. CONCLUSIONS

We presented a brief overview of our recent results in the application of memristive technology to neuromorphic computing. We started by introducing memristive devices and their crossbar integration. The CMOL circuits have been also introduced as the final goal, a way to a practical implementation of neuromorphic hardware. Successively, two fundamental experimental results have been presented. The first is experimental demonstration of SLP network based on a passive crossbar circuits. This work demonstrated the feasibility of vector-by-matrix product on the physical level, which is the building block of much bigger and more powerful ANNs. Further, we showed the capability of memristive devices to reproduce the fundamental behavior of biological synapses. This is the first step towards the realization of spiking neural networks, the most advanced and the most powerful generation of ANNs. Lastly, we discussed the issues that remain open. Unquestionably, neuromorphic computing is a multidisciplinary field that presents opportunity to overcome many limits of current computing technology. If successful, neuromorphic computing will enable practical implementations of many low cognitive tasks, such as image or speech recognition, and eventually could pave the way towards more ambitious artificial intelligence applications.

REFERENCES

- [1] W. S. McCulloch, and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," The bulletin of mathematical biophysics, 5(4), 115-133 (1943).
- [2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, 521(7553), 436-444 (2015).
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, 1097-1105 (2012).
- [4] C. Farabet, C. Couprie, L. Najman *et al.*, "Learning hierarchical features for scene labeling," Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(8), 1915-1929 (2013).
- [5] J. Ma, R. P. Sheridan, A. Liaw *et al.*, "Deep Neural Nets as a Method for Quantitative Structure-Activity Relationships," Journal of chemical information and modeling, 55(2), 263-274 (2015).
- [6] T. Ciodaro, D. Deva, J. De Seixas *et al.*, "Online particle detection with Neural Networks based on topological calorimetry information," Journal of Physics: Conference Series, 368, 012030 (2012).
- [7] H. Markram, E. Muller, S. Ramaswamy *et al.*, "Reconstruction and Simulation of Neocortical Microcircuitry," Cell, 163(2), 456-492.
- [8] K. K. Likharev, "CrossNets: Neuromorphic hybrid CMOS/nanoelectronic networks," Science of Advanced Materials, 3(3), 322-331 (2011).
- [9] L. O. Chua, and S. M. Kang, "Memristive devices and systems," Proceedings of the IEEE, 64(2), 209-223 (1976).
- [10] R. Waser, and M. Aono, "Nanoionics-based resistive switching memories," Nature materials, 6(11), 833-840 (2007).
- [11] M. D. Pickett, D. B. Strukov, J. L. Borghetti *et al.*, "Switching dynamics in titanium dioxide memristive devices," Journal of Applied Physics, 106(7), 074508 (2009).
- [12] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain *et al.*, "Memristor-based memory: The sneak paths problem and solutions," Microelectronics Journal, 44(2), 176-183 (2013).
- [13] K. K. Likharev, "Hybrid CMOS/nanoelectronic circuits: Opportunities and challenges," Journal of Nanoelectronics and Optoelectronics, 3(3), 203-230 (2008).
- [14] D. Tu, M. Liu, W. Wang *et al.*, "Three-dimensional CMOL: three-dimensional integration of CMOS/nanomaterial hybrid digital circuits," Micro & Nano Letters, IET, 2(2), 40-45 (2007).
- [15] D. B. Strukov, and K. K. Likharev, "Prospects for the development of digital CMOL circuits," 2007 IEEE-ACM International Symposium on Nanoscale Architectures (NANOARCH), 109 - 116 (2007).

- [16] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins *et al.*, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, 521(7550), 61-64 (2015).
- [17] J. Hertz, A. Krogh, and R. G. Palmer, [Introduction to the theory of neural computation] Basic Books, (1991).
- [18] W. Gerstner, R. Ritz, and J. L. Van Hemmen, "Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns," *Biological cybernetics*, 69(5-6), 503-515 (1993).
- [19] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural networks*, 10(9), 1659-1671 (1997).
- [20] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, 345(6197), 668-673 (2014).
- [21] T. Pfeil, A. Grübl, S. Jeltsch *et al.*, "Six networks on a universal neuromorphic computing substrate," *Frontiers in neuroscience*, 7, (2013).
- [22] T. Masquelier, and S. Thorpe, "Learning to recognize objects using waves of spikes and Spike Timing-Dependent Plasticity," *The 2010 International Joint Conference on Neural Networks (IJCNN)*, (2010).
- [23] F. Ponulak, and A. Kasinski, "Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting," *Neural Computation*, 22(2), 467-510 (2010).
- [24] G.-q. Bi, and M.-m. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *The Journal of neuroscience*, 18(24), 10464-10472 (1998).
- [25] W. Gerstner, R. Kempter, J. L. van Hemmen *et al.*, "A neuronal learning rule for sub-millisecond temporal coding," *Nature*, 383(LCN-ARTICLE-1996-002), 76-78 (1996).
- [26] N. Caporale, and Y. Dan, "Spike Timing-Dependent Plasticity: A Hebbian Learning Rule," *Annual Review of Neuroscience*, 31(1), 25-46 (2008).
- [27] M. Prezioso, Y. Zhong, D. Gavrilov *et al.*, "Spiking Neuromorphic Networks with Metal-Oxide Memristors", 2016 IEEE International Symposium on Circuits and Systems, (2016) – Accepted
- [28] M. Prezioso, F. Merrik-hbayat, B. D. Hoskins *et al.*, "Self-Adaptive Spike-Time-Dependent Plasticity of Metal-Oxide Memristors," *Scientific Reports*, (2016) – Accepted
- [29] J. Rubin, D. D. Lee, and H. Sompolinsky, "Equilibrium properties of temporally asymmetric Hebbian plasticity," *Physical Review Letters*, 86(2), 364 (2001).
- [30] M. C. Van Rossum, G. Q. Bi, and G. G. Turrigiano, "Stable Hebbian learning from spike timing-dependent plasticity," *The Journal of Neuroscience*, 20(23), 8812-8821 (2000).
- [31] R. Güttig, R. Aharonov, S. Rotter *et al.*, "Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity," *The Journal of neuroscience*, 23(9), 3697-3714 (2003).
- [32] S.-S. Sheu, M.-F. Chang, K.-F. Lin *et al.*, "A 4Mb embedded SLC resistive-RAM macro with 7.2 ns read-write random-access time and 160ns MLC-access capability," 2011 IEEE International Solid-State Circuits Conference, 200-202 (2011).
- [33] T.-y. Liu, T. H. Yan, R. Scheuerlein *et al.*, "A 130.7-2-Layer 32-Gb ReRAM Memory Device in 24-nm Technology," *Solid-State Circuits, IEEE Journal of*, 49(1), 140-153 (2014).
- [34] I. Kataeva, F. Merrikh-Bayat, E. Zamanidoost *et al.*, "Efficient training algorithms for neural networks based on memristive crossbar circuits," 2015 International Joint Conference on Neural Networks (IJCNN), 1-8 (2015).
- [35] Y. Huang, R. Huang, Y. Pan *et al.*, "A new dynamic selector based on the bipolar RRAM for the crossbar array application," *Electron Devices, IEEE Transactions on*, 59(8), 2277-2280 (2012).
- [36] J. Joshua Yang, M.-X. Zhang, M. D. Pickett *et al.*, "Engineering nonlinearity into memristors for passive crossbar applications," *Applied Physics Letters*, 100(11), 113501 (2012).
- [37] J. Woo, D. Lee, E. Cha *et al.*, "Vertically stacked ReRAM composed of a bidirectional selector and CB-RAM for cross-point array applications," *Electron Device Letters, IEEE*, 34(12), 1512-1514 (2013).
- [38] J. Sung Hyun, T. Kumar, M. Asnaashari *et al.*, "3D ReRAM with Field Assisted Super-Linear Threshold (FASTTM) Selector technology for super-dense, low power, low latency data storage systems," *Design Automation Conference (ASP-DAC)*, 2015 20th Asia and South Pacific, 575-575 (2015).