On Neuromorphic Spiking Architectures for Asynchronous STDP Memristive Systems

J. A. Pérez-Carrasco, C. Zamarreño-Ramos, T. Serrano-Gotarredona, and B. Linares-Barranco

Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC). C/ Américo Vespucio s/n, 41092 Sevilla, Spain. E-mail: *bernabe@imse-cnm.csic.es*

***Abstract*– Neuromorphic circuits and systems techniques have great potential for exploiting novel nanotechnology devices, which suffer from great parametric spread and high defect rate. In this paper we explore some potential ways of building neural network systems for sophisticated pattern recognition tasks using memristors. We will focus on spiking signal coding because of its energy and information coding efficiency, and concentrate on Convolutional Neural Networks because of their good scaling behavior, both in terms of number of synapses and temporal processing delay. We propose asynchronous architectures that exploit memristive synapses with specially designed neurons that allow for arbitrary scalability as well as STDP learning. We present some behavioral simulation results for small neural arrays using electrical circuit simulators, and system level spike processing results on human detection using a custom made event based simulator.**

I. Introduction

We are witnessing an explosion of nanotechnology devices in recent years with applications not only in electronics, but also combining other phenomena such as optics, chemical sensing, bio-organic based sensing, etc. which arise naturally when exploiting nano-scale dimensions. In this paper we focus mainly on the potential for building large scale computational systems, keeping the human brain as a candidate for imitation. A great variety of two-, three-, four- or more terminal computational nano-devices are being reported with potential use as resistors (memristors), or traditional FETs, which can find immediate use in traditional digital or mixed circuits and systems, but at a much larger scale. However, new nano-scale devices suffer from great parametric variations and defects, which makes necessary the use of some system level conception to overcome such drawback. Biological brains also use devices (such as neurons and synapses) which have a limited life time, are slow and present great parametric spread. However, brains circumvent such problems at the system level, through clever architectures and adaptation techniques resulting in robust and highly efficient intelligent systems. Luckily, during the past decades the engineering community has been developing circuits and systems taking inspiration from biological brains, called "neuromorphic" circuits and systems. It seems quite natural now to exploit such knowledge for designing and building large scale neuromorphic systems using nano scale devices, capable of circumventing their limitations through neuro-inspired learning. In this paper we present some viable solutions in this respect.

II. Devices

We will consider here the memristor, which is an adaptive 2-terminal resistive device. Our objective is to exploit such device as the synaptic element of a neural perceptron. Neurons can be designed using available CMOS VLSI technology, while synapses (which are required in much larger quantities) can be fabricated as nano-scale devices arranged on top of a silicon chip using some post-CMOS fabrication technique, in a CMOL-like arrangement [1]. The synaptic devices require two modes of operation: (1) a computational mode in which they contribute to a neuron's integral with a characteristic weight, and (2) an adaptation mode in which they change its characteristic weight when their terminal voltages meet some requirement. In the first mode we will use the devices as resistors, while in the second we want to change its conductance when some of their terminal voltage difference exceeds a threshold v_{th}. For example, Fig. 1 shows symbols and characteristics learning function of a voltage controlled memristor which can be defined by the following equation [2]-[3]

$$i_{MR} = G(w)v_{MR} \qquad \dot{w} = f(v_{MR}) \tag{1}$$

If $|v_{MR}| < v_{th}$ its conductance does not change, otherwise it changes according to eq. (1), where w is a parameter controlling conductance $G \in [G_{min}, G_{max}]$. In the next Section we develop a convenient memristor macro model for electric circuit simulation.

III. Memristor Macro model

Let us consider the memristor model used by DiVentra et al [4]-[6], which is a particular case of eq. (1), defined as

$$i_{MR} = (1/R)v_{MR} \quad , \quad \dot{R} = f(v_{MR}, R) \tag{2}$$

The equation for $f(v_{MR}, R)$ used by them is of the form

$$f(v_{MR}, R) = f(v_{MR})H(R) \tag{3}$$

where $f(v_{MR})$ could be described by Fig. 1(b) or a piece-wise-linear approximation [4]-[6]. Nonlinear function $H(R)$ is to guarantee that variable R remains restricted to the interval $[R_{min}, R_{max}]$. The particular function they propose is $H(R) = \theta(R - R_{min}) \times \theta(R_{max} - R)$, where $\theta(\cdot)$ is the step function. However, mathematically, this expression has the drawback that if variable R exceeds the limits $[R_{min}, R_{max}]$ at some instant, then $\dot{R}$ will stay equal to zero for ever and R will not change any more. Alternatively, one may use a smoothed version of $\theta(\cdot)$ which reduces this problem, but does not eliminate it always. Note that the objective of multiplying function $H(R)$ is to restrict the values of R to the interval $[R_{min}, R_{max}]$. Such objective can also be accomplished by substituting eq. (3) by

$$f(v_{MR}, R) = f(v_{MR}) - f_{sat}(R) \tag{4}$$

where nonlinear function $f_{sat}(R)$ is such that it is zero if $R \in [R_{min}, R_{max}]$ and grows very rapidly otherwise fully absorbing any contribution from $f(v_{MR})$, thus making $\dot{R} \approx 0$ in eq. (2). Function $f_{sat}(R)$ will thus have a shape as shown in Fig. 1(c). We will now provide a macro model circuit for implementing eqs. (2) and (4) in Spectre. A macro model of a device is a behavioral model made of circuit elements (ideal or not) that describe the same behavior. Note that some circuit simulators allow to define a device mathematically using AHDL or Verilog-A. However, if it is possible to describe it with a macro model, it will have some advantages. (1) First, it uses already built-in components providing faster simulations; (2) second, as it is made of circuit elements it gives a richer intuitive insight to (analog) circuit designers on how it works and performs, and how to improve it for specific goals; (3) it is very intuitive to add parasitic components (resistors and capacitors) to aid in the convergence of the simulator internal algorithms; (4) and if one is careful in keeping the operating voltages and currents of internal nodes to the levels the simulator

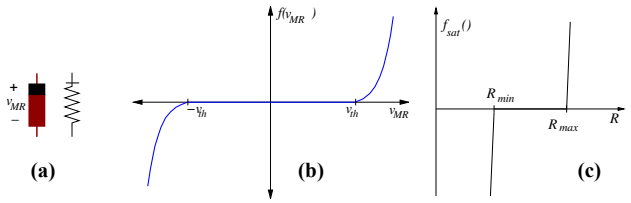

Fig. 1: Memristor (a) symbol, (b) characteristic learning function, and (c) saturating function for restricting R to the interval $[R_{min}, R_{max}]$.

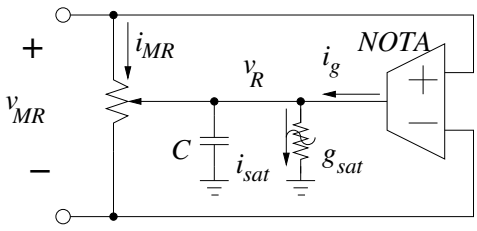

Fig. 2: Memristor macro model circuit for electric simulations

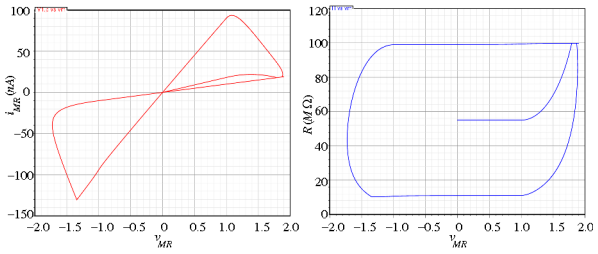

Fig. 3: Macro model based simulation of memristor. Left is memristors current and right its instantaneous resistance.

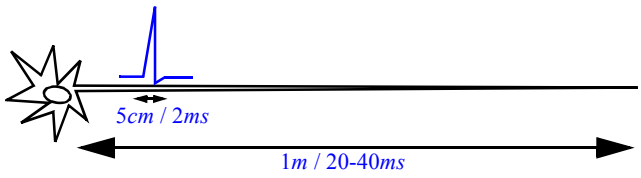

Fig. 4: biological action potential travelling through a nerve

expects from conventional circuits, simulations converge easier and faster.A circuit macro model that implements eqs. (1) and (4) is shown in Fig. 2. It is composed of a controlled resistor whose resistance R is controlled linearly by internal state voltage v_R

$$R(v_R) = k_R v_R \tag{5}$$

Component *NOTA* is a nonlinear transconductor, also known as nonlinear OTA (Operational Transconductance Amplifier) and it provides an output current $i_g(v_{MR})$ controlled by input differential voltage v_{MR}. OTA current i_g is, in our case, a piece-wise linear approximation of the function in Fig. 1(b). Nonlinear element $g_{sat}(v_R)$ is a nonlinear resistor with a piece-wise linear shape as shown in Fig. 1(c), but where R is replaced by v_R, $[R_{min}, R_{max}]$ by $[v_{Rmin}, v_{Rmax}]$, and $f_{sat}(R)$ by current $i_{sat}(v_R)$. Consequently, the macro model circuit in Fig. 2 is mathematically described by

$$\begin{aligned} v_{MR} &= R(v_R) i_{MR} \\ R(v_R) &= k_R \times (v_R + v_{R0}) \\ i_g(v_{MR}) &= C\dot{v}_R + i_{sat}(v_R, v_{Rmin}, v_{Rmax}) \end{aligned} \tag{6}$$

Parameter k_R scales between the voltage domain range of v_R (usually within a few volts, for proper simulator convergence) to the resistance domain range of R which can be as high as hundreds of Mega-ohms.Fig. 3 shows the simulation results of a memristor stimulated with a $2V$ sinusoid of $1KHz$ and with $R_{min} = 10M\Omega$, $R_{max} = 100M\Omega$.

IV. Spiking Signals

Biological brains code and transmit information as spiking signals. This is because spike coding is highly energy efficient as well as information processing efficient. Fig. 4 shows a typical $2ms$ "action potential" neural spike travelling about $1m$ from the brain to a finger muscle in about 20-40ms. In space, such spike only charges about $5cm$ of nerve, but not the whole line. The spike sender only needs to provide the charge for this $5cm$ travelling segment. This contrasts with present day electronic systems where digital information (bits) are transmitted by fully charging and later

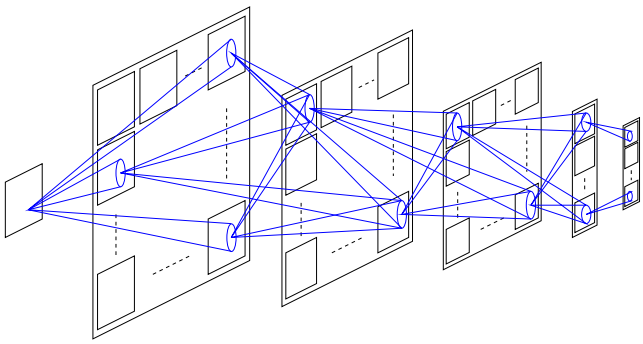

Fig. 5: Typical structure of a Convolution Neural Network (ConvNet)

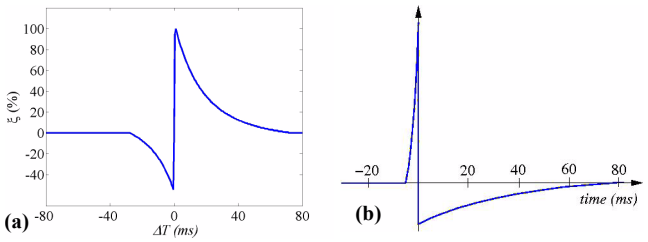

Fig. 6: (a) STDP characterization in biological synapses. Vertical axis is synaptic strength change and horizontal axis is time delay between pre- and post-synaptic spikes. (b) Action potential waveform.

discharging wires. Furthermore, when using transmission lines for high speed, there is usually a 50Ω resistive component, besides the capacitive load, which is permanently dissipating energy (even in the absence of information transmission). Spike encoding is therefore highly energy efficient and could be implemented using soliton technology [7]. Regarding information coding, spikes allow for very efficient schemes. Thorpe demonstrated in 1996 [8] that, in the human brain, fast recognition of sophisticated figures (such as animal detection in a photograph) is performed in a feed forward manner in such a way that the neurons involved only generate one spike. Thorpe later developed spike-processing convolutional architectures capable of performing this type of recognition efficiently in software [9].

V. Convolutional Neural Architectures

Convolutional neural architectures were originally proposed by Fukushima [10], taking inspiration from neuroscience [11], and later developed by LeCun [12] and used in many applications. Some researchers are proposing powerful brain models based on convolutional architectures [13]. Fig. 5 shows a typical Convolutional Neural Network architecture. They usually contain a reduced number of sequential layers (4-10), each of which performs several 2D filtering operations in parallel. Early stages extract simple features (such as edge orientation and scale), which are progressively combined into more complex shapes and figures at later stages. Early stages usually operate with small but dense kernels, while later stages use longer range but sparser ones [13]. To increase the knowledge (dictionary of shapes and figures) of the system one simply has to add more 2D filters in later layers. Example Convolutional systems for face and character recognition applications may have several tens to hundreds filters per layer. What is interesting about Convolutional Neural Networks, compared to other neural networks, is their graceful scaling capability. To increase knowledge one simply has to increase the number of modules in a layer. Number of neurons (pixels) scales linearly with the number of modules. Each module performs several filters. There is a fixed number of synapses per filter (the convolutional kernel weights). Consequently, number of synapses also scales linearly with the number of modules. On the other hand, the latency of the computing structure (if implemented as parallel hardware) is determined mainly by the number of sequential layers, which is a reduced number and does not change for a given application. In other neural network architectures the number of

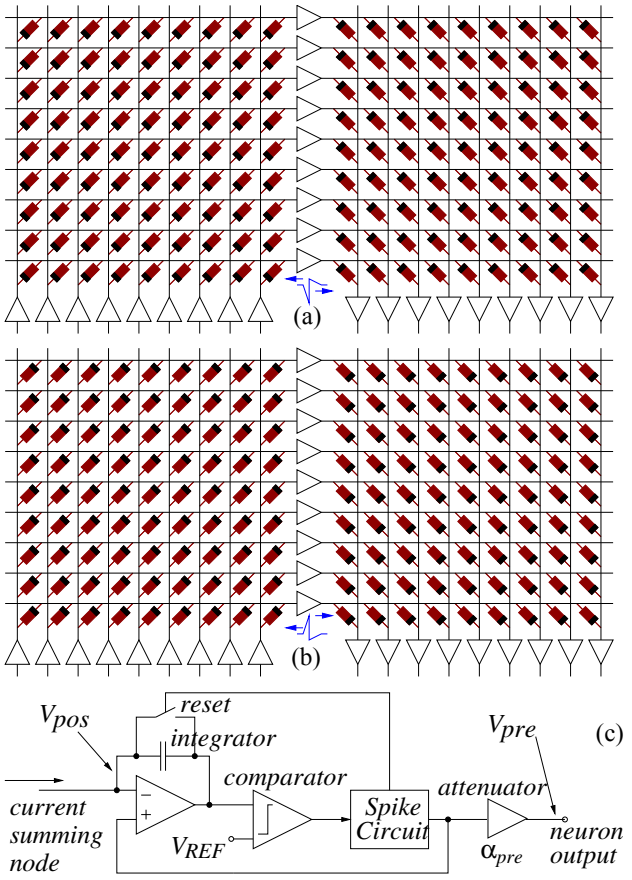

Fig. 7: (a-b) Crossbar arrangements for feed forward memristive synapse STDP neural network. (c) Conceptual block diagram of neuron.

synapses scales quadratically with the number of neurons. Consequently, Convolutional Neural Networks seem very appealing for configurable, modular and scalable (spiking or non-spiking) hardware implementations.

VI. Spike Time Dependent Plasticity

Spike-time-dependent-plasticity (STDP) is a neural learning mechanism originally postulated [14] in the context of artificial machine learning algorithms (or computational neuroscience) exploiting spike-based computations (as in brains). Astonishingly, experimental evidences of biological STDP have later been reported by several neuroscience groups worldwide [15]. In STDP the change in synaptic weight Δw is expressed as a function ξ of the time difference between the post-synaptic spike at t_{pos} and the pre-synaptic spike at t_{pre}. Specifically, $\Delta w = \xi(\Delta T)$, with $\Delta T = t_{pos} - t_{pre}$. The shape of the STDP function ξ can be interpolated from experimental data from Bi and Poo [15] as shown in Fig. 6(a). For positive ΔT there will be a potentiation of synaptic weight $\Delta w > 0$, which will be stronger as $|\Delta T|$ reduces. For negative ΔT there will be a depression of synaptic weight $\Delta w < 0$, which will be stronger as $|\Delta T|$ reduces. We recently demonstrated [16]-[17] that if a memristor (as defined in eq. (1)) is stimulated on its two terminals by two asynchronous spiking signals of the shape shown in Fig. 6(b) separated by a time ΔT, and attenuating the post-synaptic one by $\alpha_{pos}<1$, then the weight update function shown in Fig. 6(a) is mathematically obtained, which is identical to the one obtained by Bi and Poo from physiological experiments. This opens the possibility that in biological synapses there might be a memristive type of mechanism responsible for biological STDP [16]. Also, it turns out that the action potential shape strongly influences the resulting STDP function [17].

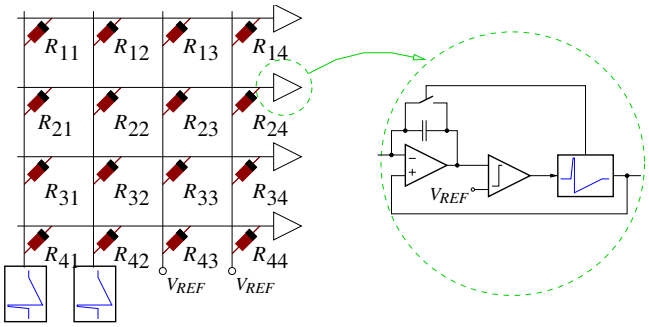

Fig. 8: Feed forward synaptic memristive array simulated behaviorally with Cadence Spectre

VII. Example Architecture

Using the result from the previous Section, we can propose a neural architecture using memristors as synapses and spiking neurons that send back a replica. Fig. 7(a-b) show two possible arrangements for implementing an STDP spiking memristive feed forward perceptron architecture. Depending on the polarity of the memristors, the neural spikes need to be inverted or not. Fig. 7(c) shows a conceptual block diagram of the neuron circuits required.

Neurons are made of integrators whose input node is maintained at virtual ground by a high-gain differential amplifier with a capacitor connected at its negative feedback input, thus acting as an integrator. At the output of the differential amplifier appears the accumulated integral (signed reversed) of the in-flowing current. Whenever this integral reaches threshold V_{REF}, an action potential spike as in Fig. 6(b) will be triggered. During the time of the action potential spike (including fast positive spike plus longer negative tail), a reset pulse will also be provided to short the integrating capacitor. This has three reasons: (a) to reset the accumulated integral, (b) to buffer the output spike to send it back through the input collecting line, (c) and to avoid any further input signal integration during spike production. An attenuated version of the spike voltage is sent forward to the next layer synapses. This architecture avoids cross-coupling of spikes between rows and columns. Using this arrangement with the memristor macro model of Fig. 2 we performed intensive behavioral simulations in Cadence-Spectre to test the concept on the 4x4 feed forward array shown in Fig. 8. The results are shown in Fig. 9. Only the first 2 column synapses are stimulated with 200*ms* period spikes (of 45*ms* duration) with a 25*ms* relative delay between the two columns. As can be seen only synapses at the first two columns change their resistance, while those on the other two columns do not, confirming the correct operation of STDP, without any crosstalk between columns nor rows. This demonstrates that this architecture can be scaled to arbitrary size, at least ideally. Practical considerations that could limit maximum size are given mainly by fan-out of neurons and interconnects delays.

VIII. People Recognition

Using the above concepts we have simulated behaviorally a spiking convolutional neural network (see Fig. 10), using a custom made simulator [19] for asynchronous event-based sensing and processing systems. The input visual flow was captured with a physical temporal contrast (motion) AER retina [20] when observing people walking. Visual pixel array was down sampled to 32x32. The spiking convolutional network has 7 layers. The first layer is a Gabor filter bank, second layer is subsampling, third layer is a trainable 5x5 kernels filter bank, fourth layer is subsampling, fifth layer is again a trainable 5x5 kernel filter bank, and sixth and seventh layers are fully connected trainable perceptrons. The system was trained off-line, in its equivalent non-spiking representation, through back propagation learning to categorize inputs as vertical humans, up side down humans, horizontal humans, or other objects. After training, learnt parameters were mapped to the spiking representation and the system was tested with new retina recordings, showing a correct recognition rate of above 86%.

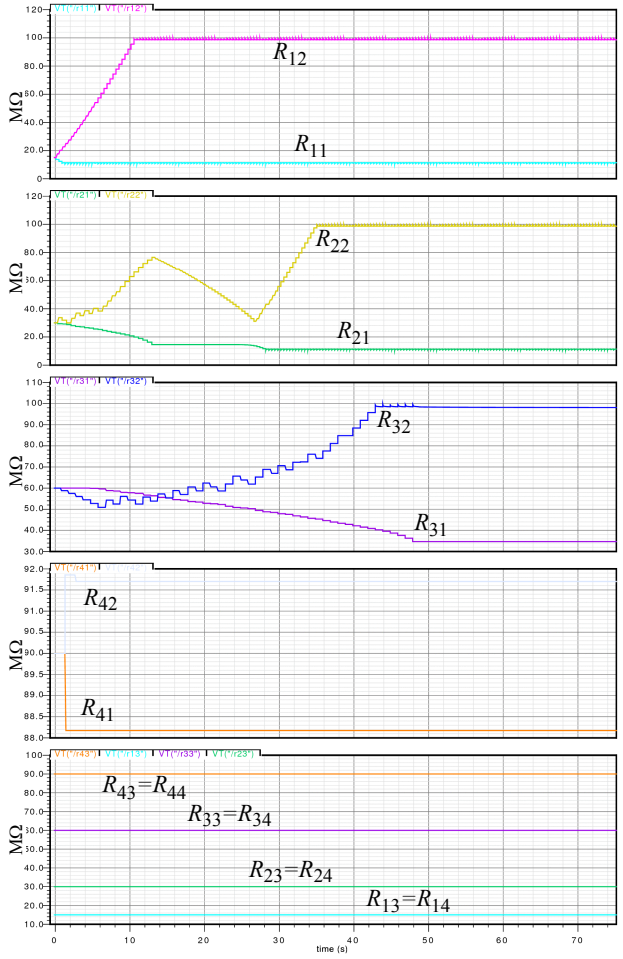

Fig. 9: Evolution of weights (resistances) of a 4x4 feed forward memristive perceptron network. Bottom trace shows the weights of memristors in the third/fourth column. The other traces show the evolution of weights in the two left most columns. Traces are grouped pair-wise with synapses in the same row (and with identical initial condition).

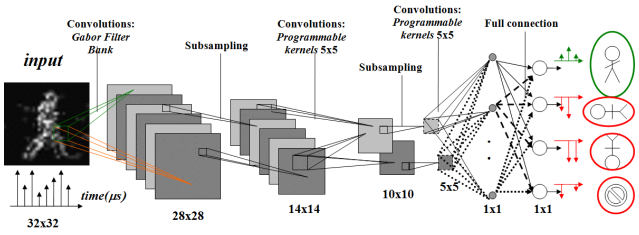

Fig. 10: Structure of spiking Convolutional Neural Network trained for recognizing people from visual data captured from an AER motion retina

IX. Conclusions and Future Outlook

We have motivated the combination of spiking neuromorphic circuit and system design with nano technology adaptive devices. We have illustrated how to implement spiking convolutional neural networks using memristors, have developed a convenient memristor macro model, and have shown simulation results of circuit structures performing correct STDP learning. We also have shown behavioral system level event-based simulations of an example application for people detection using real sensory data from an AER motion retina. Present day memristors suffer from some shortcomings that difficult the physical realization of operative STDP structures: (a) they tend to go quickly to their limit range values G_{min}, G_{max} making it difficult to set intermediate range values, and (b) in the range below threshold ($|v_{MR}| < v_{th}$) the resistance still is weakly updated. Adaptive nano-FETs show better behavior in this respect, although they are less compact. However, it is possible to conceive memristive circuits with the same behavior shown before, but using adaptive FETs as synapses [21]-[23]. Future work will target full STDP learning in spiking hardware within a complex learning system similar to those like in Fig. 10 [25].

X. Acknowledgements

This work was supported by EU grant 216777 (NABAB), Spanish grants TEC2006-11730-C03-01 (SAMANTA2) and TEC2009-10639-C04-01 (VULCANO), and Andalucian grant P06TIC01417 (Brain-System). JAPC was supported by Brain-System and CZR by an FPU scholarship.

XI. References

[1] D. B. Strukov and K. K. Likharev, "CMOL FPGA: a configurable architecture for hybrid digital circuits with two-terminal nanodevices," *Nanotechnology* 16, pp. 888-900, 2005.

[2] L. O. Chua and S. M. Kang, "Memristive Devices and Systems," *Proc. IEEE* vol. 64, No. 2, pp. 209-223, 1976.

[3] G. S. Snider, "Spike-Timing-Dependent Learning in Memristive Nanodevices," *IEEE Int. Symp. Nano Architectures,* pp. 85-92, June 2008.

[4] Pershin, Yuriy and Di Ventra, Massimiliano, "Experimental demonstration of associative memory with memristive neural networks," *Nature Precedings* http://hdl.handle.net/10101/npre.2009. 3258.1, May 19th, 2009.

[5] Pershin, Yuriy and Di Ventra, Massimiliano, "Practical approach to programmable analog circuits with memristors," http://arxiv.org/pdf/0908.3162.

[6] Y. V. Pershin, S. La Fontaine, and M. Di Ventra, "Memristive model ofamoeba.s learning," *Phys. Rev. E*, 80, 021926, 2009.

[7] D. S. Ricket, X. Li, N. Sun, K. Woo, and D. Ham, "On the Self-Generation of Electrical Soliton Pulses," *IEEE J. Solid-State Circ.,* vol. 42, No. 8, Aug. 2007.

[8] S. Thorpe, D. Fize, C. Marlot, "Speed of processing in the human visual system," *Nature* 381: 520-2, 1996.

[9] S. Thorpe, R. Guyonneau, N. Guilbaud, J.M. Allegraud, R. Vanrullen, "SpikeNet: Real-time visual processing with one spike per neuron," *Neurocomputing* 58-60: 857-64, 2004.

[10] K. Fukushima and N. Wake, "Handwritten Alphanumeric Character Recognition by the Neocognitron," *IEEE Trans. Neural Networks*, vol. 2, No. 3, pp. 355-365, May 1991.

[11] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," J. Physiol. 148, pp. 574-591, 1959.

[12] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time Series," in *The Handbook of Brain Science and Neural Networks*, M. Arbib (Ed.), Cambridge, MA: MIT Press, pp. 255-258, 1995.

[13] T. Serre, L. wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust Object Recognition with Cortex-Like Mechanisms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, No. 3, pp. 411-426, March 2007.

[14] W. Gerstner, et al. "Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns," *Biological Cybernetics,* 69, 503-515, 1993.

[15] G. Bi and M. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.,* 18(24), 10464-10472, 1998.

[16] B. Linares-Barranco and T. Serrano-Gotarredona, "Memristance can explain Spike-Time-Dependent-Plasticity in Neural Synapses," *Nature Precedings* <http://hdl.handle.net/10101/npre.2009.3010.1> 31st March, 2009.

[17] B. Linares-Barranco and T. Serrano-Gotarredona, "Exploiting Memristance in Adaptive Asynchronous Spiking Neuromorphic Nanotechnology Systems," *Proc. IEEE NANO*, July 2009.

[18] B. Linares-Barranco et al., "Compact Low-Power Calibration Mini-DACs for Neural Massive Arrays with Programmable Weights," *IEEE Trans. Neural Networks*, vol. 14, No. 5, pp. 1207-1216, September 2003.

[19] J. A. Pérez-Carrasco, et al., "On the Computational Power of AER Vision Processing Hardware," *Proc. XXII Int. Conf. Design Circ. & Integr. Systems* (DCIS), Sevilla, Spain, Nov. 2007.

[20] P. Lichtsteiner, C. Posch, and T. Delbrück, "A 128x128 120dB 30mW Asynchronous Vision Sensor that Responds to Relative Intensity Change," *IEEE J. Solid-State Circ.*, vol. 43, No. 2, pp. 566-576, Feb. 2008.

[21] S. Ramakrishna, et al. "Floating Gate Synapses with Spike Time Dependent Plasticity," *IEEE Int. Symp. Circ. & Syst.*, ISCAS 2010.

[22] N. Archontas et al. "Characterization of Memristive Poly-Si Nanowires via Empirical Physical Modelling," *IEEE Int. Symp. Circ. & Syst.*, ISCAS 2010.

[23] G. Agnus, et al. "Carbon Nanotube-based Programmable Devices for Adaptive Architectures," *IEEE Int. Symp. Circ. & Syst.*, ISCAS 2010.

[24] O. Bichler et al. "Development of a Functional Model for the Nanoparticle-Organic Memory Transistor," *IEEE Int. Symp. Circ. & Syst.*, ISCAS 2010.

[25] T. Masquelier and S. Thorpe, "Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity," Plos Comp. Biology, 3(2): e31. doi:10.1371/journal.pcbi.0030031, 2007.