Beyond Spike-Timing Dependent Plasticity in Memristor Crossbar Arrays

Hesham Mostafa*, Christian Mayr†, and Giacomo Indiveri*

*Institute of Neuroinformatics, University of Zürich and ETH Zürich, Switzerland
†Institute of Circuits and Systems, Technical University of Dresden, Germany

Email: hesham@ini.uzh.ch

***Abstract*—Memristors have emerged as promising, area-efficient, nano-scale devices for implementing models of synaptic plasticity in hybrid CMOS-memristor neuromorphic architectures. These architectures aim at reproducing the learning capabilities of biological networks by emulating the complex dynamics of biological neurons and synapses. However, to maximize the density of these elements in crossbar arrays, learning circuits have often been limited to the implementation of simple spike timing-dependent plasticity (STDP) mechanisms. We propose novel hybrid CMOS-memristor circuits that reproduce more effective and realistic plasticity rules which depend on the timing of the pre-synaptic input spike and on the state of the post-synaptic neuron, and which allow the integration of dense crossbar memristor arrays. To implement these plasticity rules in memristor crossbar arrays, the circuits driving the memristors' post-synaptic terminals actively sense the activity on the pre-synaptic terminals to apply the appropriate stimulation waveforms across the memristors. We illustrate the advantages of this scheme by using it to implement a spike-based perceptron plasticity rule.**

I. Introduction

Biological neural networks process information in a massively parallel fashion using elements that can not be clearly categorized as either computation or memory primitives. A biological synapse is not only a memory element that stores the connection weight between two neurons, it also modulates its weight in a state and activity dependent manner, making it a processing element in its own right. Memristors are also memory elements whose values/conductances change in an activity dependent manner making them prime candidates for implementing plastic synapses in neuromorphic systems [1].

Memristive crossbar arrays have been used before to implement synaptic matrices in neuromorphic systems. However, previous work has mainly considered basic STDP plasticity protocols that depend only on the timing difference between pre- and post-synaptic spikes [2], [3], or some variants thereof [4]. In these schemes, both pre- and post-synaptic spikes trigger different types of waveforms on the pre- and post-synaptic memristor terminals. Typically these waveforms are produced in such a way so that when they overlap, the voltage across the memristor is large enough to induce weight/conductance change. On the other hand, more recent computational neuroscience models of plasticity trigger weight updates on pre-synaptic spikes and calculate the weight update value depending on the internal state variables in the pre- and post-synaptic neurons [5], [6]. No clear method yet exists for implementing these rules on CMOS-memristor architectures. A hybrid CMOS-memristor system that implemented a variant of the rule described in [5] was proposed in [7]. However that system did not support crossbar operation, precluding the possibility of using dense memristor crossbar arrays. The present paper addresses this shortcoming by proposing a new scheme for implementing plasticity rules that go beyond standard STDP on memristive crossbar arrays.

II. The Plasticity Model and Memristor Properties

The plasticity rules we consider update the synaptic weight when a pre-synaptic spike occurs by an amount that depends on the state of the post-synaptic neuron: let $\mathbf{s}(t)$ be a vector denoting the state variables in the post-synaptic neuron at time t. At a pre-synaptic spike at t_{pre}, the synaptic weight X is updated according to:

$$\mathrm{X} \leftarrow \mathrm{X} + r(\mathbf{s}(t_{pre})) \tag{1}$$

where r is a function that yields the synaptic weight increment/decrement as a function of the post-synaptic neuron state.

Our goal is to realize a neuromorphic architecture in which the spiking neurons are recurrently connected using a crossbar array of memristive synaptic elements, and where the synaptic elements follow the plasticity rule in Eq. 1. The architecture has the form shown in Fig. 1a. Each neuron element (NE) comprises an integrate and fire neuron circuit, such as the neuron in [8]; a plasticity circuit that has access to the neuron state and calculates the plasticity function $r(\mathbf{s}(t))$; and pre- and post-synaptic interface circuits that apply the appropriate voltage waveforms across the memristor elements based on $r(\mathbf{s}(t))$ in order to modulate their conductance/synaptic weights according to Eq. 1. When a neuron spikes, the post-synaptic interface circuits also inject current into each post-synaptic neuron that is proportional to the conductance of the memristor (synaptic weight) connecting it to the spiking neuron.

We consider memristors whose i-v characteristic can be generically described as:

$$i(t) = g(t)v(t) \tag{2a}$$

$$\frac{dg(t)}{dt} = \begin{cases} K & \text{if } g(t) < g_{min} \\ -K & \text{if } g(t) > g_{max} \\ f(v(t)) & \text{otherwise} \end{cases} \tag{2b}$$

978-1-4799-5341-7/16/$31.00 ©2016 IEEE

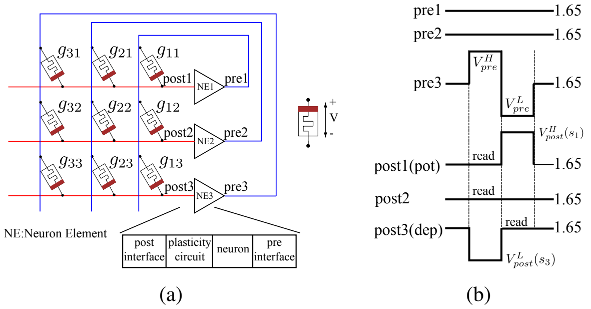

Fig. 1: (a) General form of the hybrid CMOS-memristor architecture, shown here for only three neurons as an illustration. Each crossbar element is a memristor/synapse that should follow the plasticity rule in Eq. 1. Also shown is the positive voltage direction used in Eq. 2. (b) Voltage waveforms applied on the 3 pre-synaptic and 3 post-synaptic terminals of the crossbar configuration in a when neuron 3 spikes. When neuron 3 spikes, Neuron 1 is in state $\mathbf{s_1}$ that causes it to potentiate its incoming synapses on pre-synaptic spikes, i.e, $r(\mathbf{s_1}) > 0$ (see Eq. 1). For neurons 2 and 3, $r(\mathbf{s_2}) = 0$ and $r(\mathbf{s_3}) < 0$ respectively.

where K is a positive constant and $g(t)$ is the memristor conductance that changes between g_{min} and g_{max} and whose time evolution depends on $v(t)$ through the function f. The memristor conductance represents the synaptic weight. f should exhibit a threshold effect, i.e, $f(v) = 0$ when $|v| < V_{th}$ so that the memristor conductance does not change when voltage pulses of amplitude less than V_{th} are applied. The conductance increases ($f(v)$ is positive) when $v > V_{th}$ and decreases ($f(v)$ is negative) when $v < -V_{th}$. The memristor model in Eq. 2 is quite general and can reproduce the popular “moving wall” model [9]. The bipolar switching characteristics and the threshold effect are compatible with a wide class of physical memristors [10], [11].

III. Pre- and Post-synaptic Voltage Waveforms

Figure 1b shows the voltage waveforms applied by the pre- and post-synaptic interface circuits across the memristor terminals. In the absence of spiking activity, all terminals are held at the middle of the supply range: 1.65V (all circuits are based on a 3.3V, 0.35um CMOS technology). When neuron 3 spikes, a bi-phasic voltage pulse between V^H_{pre} and V^L_{pre} is applied on terminal pre3. These voltage levels are chosen so that $V^H_{pre} - 1.65 < V_{th}$ and $1.65 - V^L_{pre} < V_{th}$ where V_{th} is the minimum voltage needed to change the conductance of a memristor. The post-synaptic interface circuits detect the perturbation caused by the bi-phasic pre-synaptic pulse and each responds in one of the following ways depending on the post-synaptic neuron state:

a) The post-synaptic neuron is in ’potentiate mode’, i.e, $r(\mathbf{s}) > 0$, such as neuron 1: The post-synaptic interface circuit clamps the post-synaptic terminal at $1.65V$ and reads the current flowing through it during the high phase of the pre-synaptic pulse. A proportional current is injected into the neuron. This synaptic current is thus proportional to the conductance of the memristor connecting to the spiking pre-synaptic neuron (g_{31} in this case). During the low phase of the pre-synaptic pulse, the post-synaptic terminal is clamped to $V^H_{post}(\mathbf{s_1})$. The voltage difference across the memristor connecting to the spiking pre-synaptic neuron is $V^H_{post}(\mathbf{s_1}) - V^L_{pre}$ and this is high enough to trigger a conductance increase proportional to $r(\mathbf{s_1})$. $V^H_{post}(\mathbf{s_1}) - 1.65 < V_{th}$ so the conductances of the other memristors are unchanged.

b) The post-synaptic neuron is in ’depression mode’, i.e, $r(\mathbf{s}) < 0$, such as neuron 3: Compared to the previous case, the memristor conductance is instead read during the low-phase of the bi-phasic pre-synaptic pulse. During the high-phase, the post-synaptic terminal is clamped at $V^L_{post}(\mathbf{s_3})$ where $1.65 - V^L_{post}(\mathbf{s_3}) < V_{th}$ and the negative voltage across memristor (3,3): $V^L_{post}(\mathbf{s_3}) - V^H_{pre}$ triggers a negative conductance/weight change proportional to $r(\mathbf{s_3})$.

c) The post-synaptic neuron is in ’neutral mode’, i.e, $r(\mathbf{s}) = 0$, such as neuron 2: The post-synaptic terminal is clamped to 1.65V throughout. Current going through the clamped post-synaptic terminal, which is proportional to the conductance g_{32}, is read during the high-phase of the pre-synaptic pulse and a proportional current is injected into the neuron.

In summary, during the bi-phasic pulse on terminal pre3 triggered by the spike of neuron 3, currents proportional to g_{31}, g_{32}, and g_{33} are injected into neurons 1, 2, and 3 respectively. Memristor (3,1) gets potentiated (g_{31} increases) and memristor (3,3) gets depressed, while memristor 3,2 stays unchanged.

IV. The Perceptron Learning Rule and Physical Implementation

We implemented the CMOS part of the architecture in Fig. 1a on an 0.35um CMOS process (the chip is currently being fabricated). In each neuron block, we used a neuron circuit that is based on the adaptive exponential integrate and fire neuron described in [8]. The plasticity model we use is based on the spike-based perceptron learning algorithm [5]. In this model, the relevant neuron states are $\mathbf{s}(t) := (V_{mem}(t), C(t))$ which are the neuron’s membrane potential and the intracellular calcium concentration, respectively. The latter is a low-pass-filtered version of the neuron’s spike train:

$$\frac{d\mathrm{C(t)}}{dt} = -\frac{1}{\tau_{\mathrm{C}}}\mathrm{C(t)} + J_{\mathrm{C}}\sum_i \delta(t - t_i) \tag{3}$$

where J_{C} and τ_{C} are the magnitude of the calcium influx during each spike and the time constant respectively, and t_i is the time of the i^{th} spike emitted by the neuron. The synaptic weight update function r (see Eq. 1) is given by:

$$r(\mathbf{s}(t)) = \begin{cases} a & \text{if } V_{mem}(t) > \theta_V \text{ and } \theta^l_{up} < C(t) < \theta^h_{up} \\ -b & \text{if } V_{mem}(t) \leq \theta_V \text{ and } \theta^l_{down} < C(t) < \theta^h_{down} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where θ_V, θ^l_{up}, θ^h_{up}, θ^l_{down}, and θ^h_{down} are the parameters of the plasticity rule and a and b are positive constants (see [7] for more details regarding the plasticity rule). The ’plasticity circuit’ block in the neuron element (see Fig. 1a) evaluates which of the three conditions in Eq. 4 holds (the implementation details of this block are given in [8]) and communicates the result to the post-synaptic interface block so that the latter can apply the correct waveforms on the post-synaptic terminal in response to bi-phasic pulses on the pre-synaptic terminals. We now describe the circuit implementations of the pre- and post-synaptic interface circuits.

A. The Pre-synaptic Interface Circuit

The pre-synaptic interface circuit should generate a biphasic pulse with adjustable duration and amplitude whenever the neuron spikes. It employs the pulse extender circuit shown in Fig. 2a. When the neuron spikes, the circuit receives a short voltage pulse on port spike which causes the capacitor to discharge to ground. The capacitor is then slowly charged to V_{dd} using a sub-threshold current passing through transistor $M1$. The bias leak_bias! which controls this current controls the capacitor charging time, and thus controls the duration of the complementary pulses generated on ports pulse and pulse_n. To have fast switching in the first inverter stage (M5-M9), a positive feedback loop is employed (M3 and M4) to pull the capacitor node quickly to V_{dd} when it approaches the switching threshold of the first inverter. Faster switching reduces power consumption as it reduces the time the inverter spends at the switching threshold. This reduces power dissipation due to cross-currents from V_{dd} to ground.

Figure 2b shows the pre-synaptic interface circuit. It employs two pulse extenders and a 1 bit digital memory implemented using the cross-coupled inverters M4-M5 and M6-M7. M4-M5 are weak transistors so the memory bit mem can be set or reset by activating the pull-down branch M1-M2 or the pull-up branch M3 respectively. Figure 2c shows transistor-level simulation results of the pre-synaptic interface circuit obtained using Cadence Spectre. Initially the digital memory, mem, is at logic 0. When the neuron spikes at 500 us, a negative pulse is generated on pulse_n1. This activates the pull-down branch M1-M2 which sets the 1 bit memory, mem, to 1. At an input spike, both inputs of the AND gate flip. The output of the AND gate, however, will not glitch as pulse_n1 goes to zero at the AND gate input before mem goes high (mem lags pulse_n1 by approximately three inverter delays). When pulse_n1 goes high (the pulse generated by PE1 is complete), the output of the AND gate goes high and this triggers a negative pulse on pulse_n2. When pulse_n2 is low, this immediately sets mem to logic 0. The second pulse then runs its course and the circuit returns to its initial state.

The 'clamp control' block is a digital memory-less block (which has 42 transistors) that uses pulse_n1 and pulse_n2 to generate the digital signals driving the gates of the wide transistors M8-M11. These transistors drive the pre-synaptic memristor terminal. Extra logic in the 'clamp control' block ensures there are no glitches can cause the pre-synaptic terminal to be simultaneously pulled towards different voltages.

B. The Post-synaptic Interface Circuit

Figure 3a shows a simplified schematic of the post-synaptic interface circuit. The circuit controls the voltage of the post-synaptic terminal through the two wide transistors M1 and M2. Each of these transistors is part of a negative feedback loop. Each feedback loop involves a high-gain amplifier (A1 or A2) that drives the gate of the wide transistor (M1 or M2) so that the voltage of the post-synaptic terminal is equal to the voltage at the negative terminal of the amplifier. The feedback loop involving M1 (M2) could thus be used to pull up (pull down) the post-synaptic terminal to a specific voltage. This voltage is selected using switches S1-S6. All switches (S1-S10) are controlled by the digital outputs of the 'switch control' block (which has 30 transistors). The 'switch control' block receives two signals: up and dn from the 'plasticity circuit' block (see Fig. 1a). These two signals indicate which of the three conditions in Eq. 4 holds.

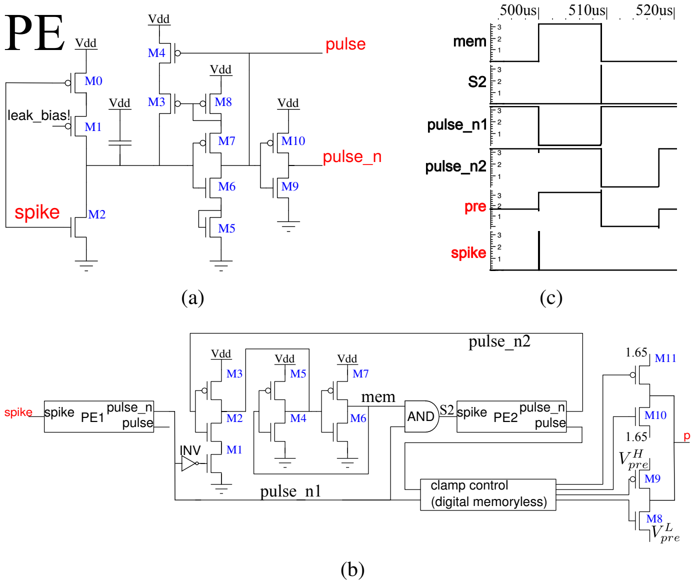

Fig. 2: The pre-synaptic interface circuit. (a) Pulse extender (PE) circuit generates 2 complementary pulses: pulse and pulse_n with adjustable duration for each spike on its spike input port. (b) Implementation of the pre-synaptic interface circuit that uses two pulse extenders and 1 bit digital memory to generate a bi-phasic pulse on output terminal pre. (c) Transient transistor-level simulation results of the circuit in b when a spike occurs at 500 us

Figure 3b illustrates the case when the post-synaptic circuit is in potentiating mode (first condition in Eq. 4). Initially, the pre-synaptic terminal is at 1.65 V and switches S2 and S5 are closed and all other switches are open. The pull-down (pull-up) path is trying to pull post to 1.8 V (1.5 V). This will cause the gates of M1 and M2 to go almost to 3.3 V and 0 V respectively, as the post-synaptic terminal voltage (1.65 V) is above the required pull-up voltage and below the required pull-down voltage. The post-synaptic terminal is thus virtually floating. When the pre-synaptic terminal goes high (to V_{pre}^{H}), drive_dn rises to try to clamp post to 1.8 V. The output of A4 (preHigh_n) therefore goes low. This is detected by the 'switch control' block which switches the input to A2 so that the pull-down path clamps post to 1.65 and closes switch S9 to disable the pull-up path. It also closes S8 so that a current proportional to the memristor current is injected into the differential pair integrator (DPI) circuit [12] (M7-M9). The DPI circuit scales and low-pass filters this current to generate the synaptic current $Isyn$ which is injected into the post-synaptic neuron.

The output of the pulse extender PE2, preHighDelay_n, eventually goes down, and in response the 'switch control' block changes the pull-down clamp voltage to 1.8 V and opens S8 to stop the current going into the synapse circuit. When the pre-synaptic terminal goes low (to V_{pre}^{L}), drive_dn decreases below 0.1 V and in response, preHigh_n and preHighDelay_n go high. This causes the 'switch control' block to open S9. drive_up will then drop to clamp post at 1.5V. This causes the output of A3 (preLow_n) to go low and in response, the 'switch control' block changes the pull-up clamp to V_{post}^{H} (3.2 V) in

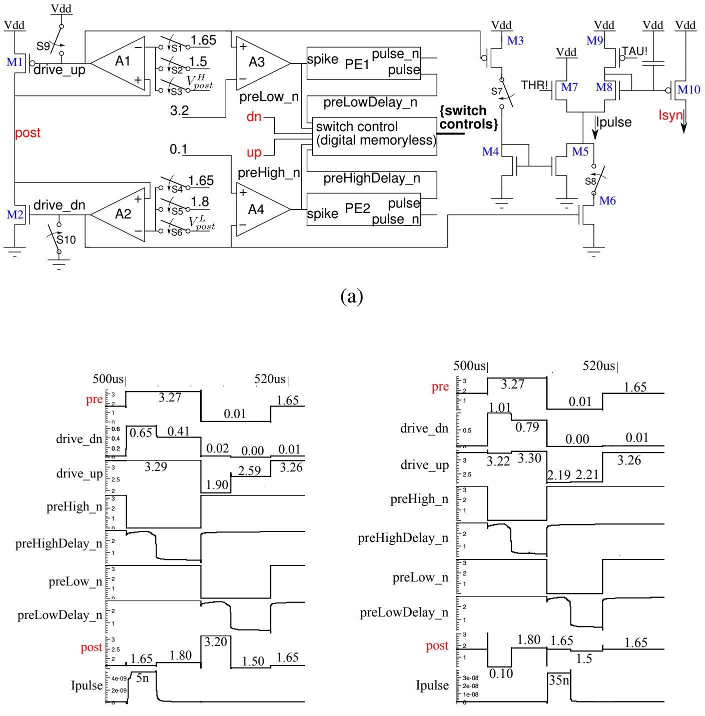

Fig. 3: The post-synaptic interface circuit. (a) Simplified schematic. (b) Transistor-level simulation results of the circuit in a in response to a bi-phasic pulse on a pre-synaptic terminal that is connected through a 100 $KOhm$ resistor to the post-synaptic terminal. Post-synaptic neuron is in potentiation mode. (c) Same as b but the post-synaptic neuron is in depression mode and the resistance connecting the pre- to post-synaptic terminals is 10 $KOhm$.

order to potentiate the memristor and closes S10 to disable the pull-down path. preLowDelay_n eventually goes down and the 'switch control' block sets the pull-up clamp voltage to 1.5 V. The pre-synaptic bi-phasic pulse completes and the pre-synaptic terminal goes back to 1.65 V. This causes drive_up to go above 3.2 V and in response preLow_n and preLowDelay_n go high. This causes the 'switch control' block to open S10 and the circuit returns to its initial state.

Figure 3c shows simulation results when the post-synaptic neuron is in depression mode. This is communicated to the 'switch control' block through the up and dn signals. The 'switch control' block clamps the post-synaptic terminal to V^L_{post} (0.1 V) during the high phase of the pre-synaptic pulse, and reads the memristor current when post is clamped to 1.65 V during the low phase. When the post-synaptic neuron is not in depression nor in potentiation mode, the 'switch control' block clamps post to 1.65 V during the high pre-synaptic phase and reads the memristor current. During the rest of the pre-synaptic pulse, it clamps post to 1.8 V and 1.5 V when the pre terminal is high or low respectively . This keeps the voltage across the memristor below 1.5 V to avoid changing its value. This is slightly better than the approach in Fig. 1b where post is clamped to 1.65 V irrespective of the pre terminal voltage.

In a 3.3 V, 0.35um CMOS process, the combined simulation static power dissipation of one pre- and one post-synaptic interface circuit is 220 nW. The circuits consume 90 nJ per spike when connected through a 10 $KOhm$ resistor and when the post-synaptic terminal is in the potentiate or depress mode, and when the duration of the biphasic pre-synaptic pulse is 160 μs. They occupy a silicon area of 4900 μm^2. The integrate and fire neuron and the plasticity circuit occupy 2200 μm^2.

V. Conclusions

We presented a novel scheme for implementing synaptic plasticity in recurrently connected spiking networks using a memristive crossbar array as the synaptic matrix. In this scheme, the post-synaptic side actively senses the state of the pre-synaptic terminals to appropriately stimulate the memristors and implement plasticity rules beyond standard STDP. We presented a concrete circuit implementation (currently being fabricated) of this scheme. Our results show how a new class of plasticity rules could be mapped to scalable CMOS-memristive neuromorphic systems that could then be used to emulate more biologically realistic learning mechanisms.

Acknowledgments

This work is partly supported by the European Union 7th framework program, project 'RAMP', (grant no. 612058) and by the EPSRC project 'CHIST-ERA' ERA-Net, EPSRC EP/J00801X/1, EP/K017829/1.

References

[1] S. Saighi, C. Mayr, B. Linares-Barranco, T. Serrano-Gotarredona, H. Schmidt, G. Lecerf, J. Tomas, J. Grollier, S. Boyn, F. Alibart, S. La Barbera, D. Vuillaume, O. Bichler, C. Gamrat, A. Vincent, and D. Querlioz, "Plasticity in memristive devices," *Frontiers in Neuroscience*, vol. 9, no. 51, 2015.

[2] C. Zamarreño-Ramos, L. Camuñas-Mesa, J. Pérez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, and B. Linares-Barranco, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Frontiers in neuroscience*, vol. 5, 2011.

[3] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, "STDP and STDP variations with memristors for spiking neuromorphic learning systems," *Frontiers in Neuroscience*, vol. 7, no. 2, 2013.

[4] C. Mayr, P. Stärke, J. Partzsch, L. Cederstroem, R. Schüffny, Y. Shuai, N. Du, and H. Schmidt, "Waveform driven plasticity in BiFeO3 memristive devices: Model and implementation," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1700–1708.

[5] J. Brader, W. Senn, and S. Fusi, "Learning real world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Computation*, vol. 19, pp. 2881–2912, 2007.

[6] C. Clopath and W. Gerstner, "Voltage and spike timing interact in STDP – a unified model," *Frontiers in Synaptic Neuroscience*, vol. 2, no. 25, pp. 1–11, 2010.

[7] H. Mostafa, A. Khiat, A. Serb, C. G. Mayr, G. Indiveri, and T. Prodromakis, "Implementation of a spike-based perceptron learning rule using TiO2x memristors," *Frontiers in Neuroscience*, vol. 9, no. 357, 2015.

[8] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A re-configurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers in Neuroscience*, vol. 9, no. 141, 2015.

[9] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[10] T. Prodromakis, B. Peh, C. Papavassiliou, and C. Toumazou, "A versatile memristor model with nonlinear dopant kinetics," *Electron Devices, IEEE Transactions on*, vol. 58, no. 9, pp. 3099–3105, 2011.

[11] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[12] C. Bartolozzi and G. Indiveri, "Synaptic dynamics in analog VLSI," *Neural Computation*, vol. 19, no. 10, pp. 2581–2603, Oct 2007.