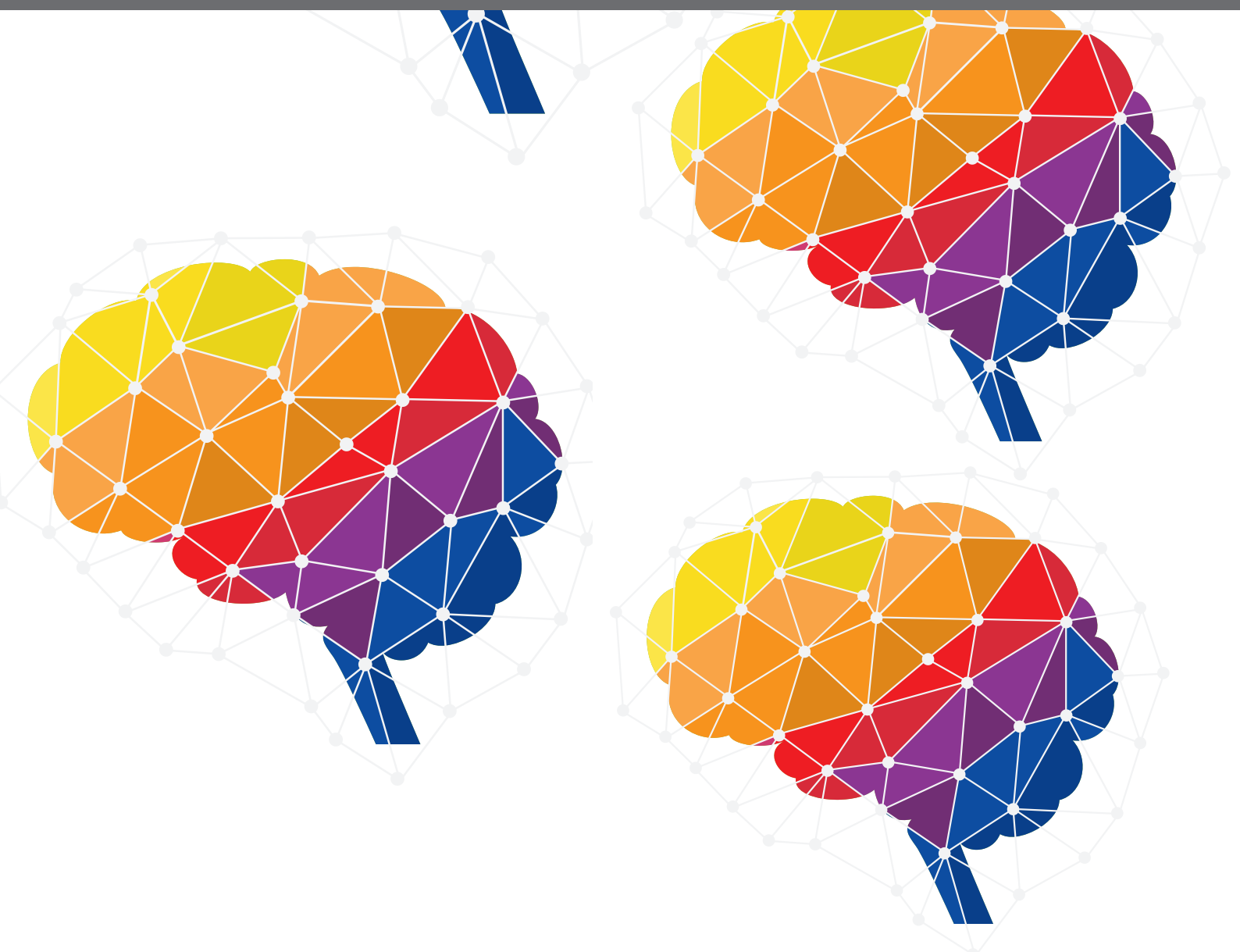
ENABLING TECHNOLOGIES FOR VERY LARGE-SCALE SYNAPTIC ELECTRONICS

EDITED BY: Themis Prodromakis and Alexantrou Serb
PUBLISHED IN: Frontiers in Neuroscience



frontiers Research Topics

frontiers

ISSN 1664-8714
ISBN 978-2-88945-508-9
DOI 10.3389/978-2-88945-508-9

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: researchtopics@frontiersin.org

ENABLING TECHNOLOGIES FOR VERY LARGE-SCALE SYNAPTIC ELECTRONICS

Topic Editors:
Themis Prodromakis, University of Southampton, United Kingdom
Alexantrou Serb, University of Southampton, United Kingdom

An important part of the colossal effort associated with the understanding of the brain involves using electronics hardware technology in order to reproduce biological behavior in 'silico'. The idea revolves around leveraging decades of experience in the electronics industry as well as new biological findings that are employed towards reproducing key behaviors of fundamental elements of the brain (notably neurons and synapses) at far greater speed-scale products than any software-only implementation can achieve for the given level of modelling detail.

So far, the field of neuromorphic engineering has proven itself as a major source of innovation towards the 'silicon brain' goal, with the methods employed by its community largely focused on circuit design (analogue, digital and mixed signal) and standard, commercial, Complementary Metal-Oxide Silicon (CMOS) technology as the preferred `tools of choice' when trying to simulate or emulate biological behavior. However, alongside the circuit-oriented sector of the community there exists another community developing new electronic technologies with the express aim of creating advanced devices, beyond the capabilities of CMOS, that can intrinsically simulate neuron- or synapse-like behavior. A notable example concerns nanoelectronic devices responding to well-defined input signals by suitably changing their internal state ('weight'), thereby exhibiting `synapse-like' plasticity. This is in stark contrast to circuit-oriented approaches where the `synaptic weight' variable has to be first stored, typically as charge on a capacitor or digitally, and then appropriately changed via complicated circuitry.

The shift of very much complexity from circuitry to devices could potentially be a major enabling factor for very-large scale `synaptic electronics', particularly if the new devices can be operated at much lower power budgets than their corresponding 'traditional' circuit replacements. To bring this promise to fruition, synergy between the well-established practices of the circuit-oriented approach and the vastness of possibilities opened by the advent of novel nanoelectronic devices with rich internal dynamics is absolutely essential and will create the opportunity for radical innovation in both fields. The result of such synergy can be of potentially staggering impact to the progress of our efforts to both simulate the brain and ultimately understand it.

In this Research Topic, we wish to provide an overview of what constitutes state-of-the-art in terms of enabling technologies for very large scale synaptic electronics, with particular stress on innovative nanoelectronic devices and circuit/system design techniques that can facilitate the development of very large scale brain-inspired electronic systems

Citation: Prodromakis, T., Serb, A., eds. (2018). Enabling Technologies for Very Large-Scale Synaptic Electronics. Lausanne: Frontiers Media.
doi: 10.3389/978-2-88945-508-9

Table of Contents

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 25 October 2016
doi: 10.3389/fnins.2016.00482

Analog Memristive Synapse in Spiking Networks Implementing Unsupervised Learning

Erika Covi[1], Stefano Brivio[1], Alexander Serb[2], Themis Prodromakis[2], Marco Fanciulli[1,3] and Sabina Spiga[1]**

[1] Laboratorio MDM, Istituto per la Microelettronica e i Microsistemi - Consiglio Nazionale delle Ricerch (CNR), Agrate Brianza, Italy, [2] Nano Group, Department of Electronics and Computer Science, University of Southampton, UK, [3] Dipartimento di Scienza Dei Materiali, Università di Milano Bicocca, Milano, MI, Italy

Emerging brain-inspired architectures call for devices that can emulate the functionality of biological synapses in order to implement new efficient computational schemes able to solve ill-posed problems. Various devices and solutions are still under investigation and, in this respect, a challenge is opened to the researchers in the field. Indeed, the optimal candidate is a device able to reproduce the complete functionality of a synapse, i.e., the typical synaptic process underlying learning in biological systems (activity-dependent synaptic plasticity). This implies a device able to change its resistance (synaptic strength, or weight) upon proper electrical stimuli (synaptic activity) and showing several stable resistive states throughout its dynamic range (analog behavior). Moreover, it should be able to perform spike timing dependent plasticity (STDP), an associative homosynaptic plasticity learning rule based on the delay time between the two firing neurons the synapse is connected to. This rule is a fundamental learning protocol in state-of-art networks, because it allows unsupervised learning. Notwithstanding this fact, STDP-based unsupervised learning has been proposed several times mainly for binary synapses rather than multilevel synapses composed of many binary memristors. This paper proposes an HfO_2-based analog memristor as a synaptic element which performs STDP within a small spiking neuromorphic network operating unsupervised learning for character recognition. The trained network is able to recognize five characters even in case incomplete or noisy images are displayed and it is robust to a device-to-device variability of up to ±30%.

Keywords: memristor, resistive switching, HfO_2, artificial synapse, synaptic plasticity, spike time dependent plasticity, spiking neuromorphic network, unsupervised learning

OPEN ACCESS

Edited by:
Gert Cauwenberghs, University of California, San Diego, USA

Reviewed by:
Siddharth Joshi, University of California, San Diego, USA
Khaled Nabil Salama, King Abdullah University of Science and Technology, Saudi Arabia

****Correspondence:***
Erika Covi
erika.covi@mdm.imm.cnr.it
Sabina Spiga
sabina.spiga@mdm.imm.cnr.it

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *16 May 2016*
Accepted: *07 October 2016*
Published: *25 October 2016*

Citation:
Covi E, Brivio S, Serb A, Prodromakis T, Fanciulli M and Spiga S (2016) Analog Memristive Synapse in Spiking Networks Implementing Unsupervised Learning. Front. Neurosci. 10:482. doi: 10.3389/fnins.2016.00482

1. INTRODUCTION

The human brain is a massively parallel, fault-tolerant, adaptive system integrating storage and computation (Kuzum et al., 2013; Matveyev et al., 2015). Moreover, it is able to visually recognize a large amount of living beings and objects and to process huge volumes of data in real-time (Kuzum et al., 2013; Yu et al., 2013a; Wang et al., 2015). Therefore, biologically-inspired systems are attracting a lot of interest as vehicles toward the implementation of real-time adaptive systems for a variety of applications. In such applications, the system is required to continuously adapt to time-varying external stimuli in an autonomous way, therefore an on-line learning without external supervision is preferable (Serb et al., 2016). In neuromorphic hardware, learning is obtained

through reconfiguration of the connectivity of a network through local modulation of synaptic weights. The adjustment of the weight of a single synapse, i.e., plasticity, should follow simple update rules that can be implemented uniformly across the entire network and allow unsupervised learning. In this respect, spike timing dependent plasticity (STDP) has been recognized as one of most promising, because it establishes that the weight of a synapse is adjusted according to the timing of the spikes fired by connected neurons (Serrano-Gotarredona et al., 2013; Bill and Legenstein, 2014; Ambrogio et al., 2016b).

Recently, the implementation of artificial synapses with memristor devices has been proposed. Memristors (memory + resistor) are compact two terminal devices that change their resistance when subjected to voltage stimulation. The memristor resistance state can be considered inversely proportional to the synaptic weight. Various practical implementations have been proposed, such as phase change (Kuzum et al., 2012; Ambrogio et al., 2016b), ferroelectric (Du et al., 2015; Nishitani et al., 2015), spin transfer torque (Querlioz et al., 2015) devices, and oxide-based resistive switching memristors (Wang et al., 2015; Ambrogio et al., 2016a). When memristors are employed in neuromorphic networks, two main operational modes are used, binary and analog. The former relies on memristors featuring only two states, high resistance state (HRS) or low resistance state (LRS), and it is proved to be effective in specific applications (Suri et al., 2013; Wang et al., 2015; Ambrogio et al., 2016a). On the other hand, analog evolution of device resistance is desirable to improve the robustness of the network (Bill and Legenstein, 2014; Garbin et al., 2015; Park et al., 2015), but the difficulty of operating memristors in an analog fashion renders hardware implementations of networks with analog synapses still challenging (Garbin et al., 2015). Indeed, several memristors show only a partial analog behavior, either when increasing the resistance (synaptic depression), which is common in filamentary devices as oxide-based memristors (Kuzum et al., 2013; Yu et al., 2013a), or when decreasing the resistance (synaptic potentiation) as in some kinds of phase change memristors (Eryilmaz et al., 2014). Well established protocols to obtain analog behavior require controlling of the current flow through the memristor (Yu et al., 2011; Ambrogio et al., 2013), or the modulation of either the time width (Park et al., 2013; Mandal et al., 2014) or the voltage (Kuzum et al., 2012; Park et al., 2013) of the spike. However, this device programming requires the use of extra circuit elements for monitoring the state of the memristor and shaping the spike accordingly. A second proposed approach is to consider multi-memristor synapses (compound synapse with stochastic programming) (Bill and Legenstein, 2014; Burr et al., 2015; Garbin et al., 2015; Prezioso et al., 2015) at the expense of increased area consumption. Only recently some works demonstrated analog behavior in both potentiation and depression without current or voltage control (Park et al., 2013; Covi et al., 2015, 2016; Matveyev et al., 2015; Brivio et al., 2016; Serb et al., 2016).

Within this class of devices, unsupervised learning based on STDP has been successfully demonstrated and analyzed in detail for binary synapses or compound synapses (with binary memristors) (Suri et al., 2013; Bill and Legenstein, 2014; Ambrogio et al., 2016a,b). Some works deal with networks utilizing analog resistance transition in only one direction, either in depression (Yu et al., 2013b) or in potentiation (Eryilmaz et al., 2014). Only few works use analog synapses to simulate neuromorphic networks, as an example Querlioz et al. (2013), Yu et al. (2015), and Serb et al. (2016). The latter, in particular, proposes a network realized in part with real hardware analog memristors and in part with software simulation.

In this framework, we propose a fully analog oxide-filamentary device as a memristive synapse for networks with deterministic neurons implementing unsupervised learning. The proposed memristor features an analog modulation of its resistance in various long-term functional plasticity spiking conditions and it emulates a type of homosynaptic STDP learning rule. To prove its usefulness in deterministic STDP-based networks, a simple fully-connected spiking neuromorphic network (SNN) for pattern recognition is conceived and simulated. The SNN consists of 30 neurons (25 pre-neurons disposed in a 5×5 layer and 5 post-neurons) and 125 synapses. The network is trained with an associative unsupervised STDP-based learning protocol. After training, the SNN is able to recognize five characters displayed as 5×5 black-and-white pixels images even when incomplete characters or noisy ones (intended as purely additive noise) are displayed. Moreover, the SNN is proved to be robust against device-to-device variability.

2. MATERIALS AND METHODS

The device stack is made of 40 nm TiN/5 nm HfO_2/10 nm Ti/40 nm TiN layers and the area of the device is $40 \times 40\ \mu m^2$. Ti and TiN layers are deposited by magnetron sputtering and the HfO_2 layer is deposited by atomic layer deposition at 300 °C, as described elsewhere (Brivio et al., 2015; Frascaroli et al., 2015). The switching mechanism of the proposed memristor is filamentary (Brivio et al., 2014), i.e., it is based on the disruption and the restoration of a conductive filament formed inside the oxide.

The electrical DC characterizations are performed using Source Measuring Units (B1511B and B1517A) of a B1500A Semiconductor Device Parameter Analyzer by Keysight. **Figure 1** shows a typical I-V curve of the device. In its pristine state, the device has a conductance of tens of nS (not shown). A forming operation (DC current sweep up to 150 μA) at around 1.8 V (data not shown) is needed to bring the device in its LRS for the first time. To switch the device from LRS to HRS, and vice versa, DC sweeps from 0 V to 1 V (LRS to HRS) and from 0 V to −0.7 V (HRS to LRS) are applied. The device maximum resistance (read at 100 mV) ratio obtainable in DC is about one order of magnitude, which is in agreement with the literature (Garbin et al., 2015; Matveyev et al., 2015; Wang et al., 2015).

The device response to spike stimulation has been characterized either by trains of pulses with increasing amplitude and fixed time width or by repetition of the same spike. In the former case, during depression spike amplitude ranges from 0.1 to 1.2 V, during potentiation, from −0.1 to −0.65 V. The same spike is repeated 5 times before the amplitude is incremented by 50 mV (decremented by −50 mV for negative voltages)

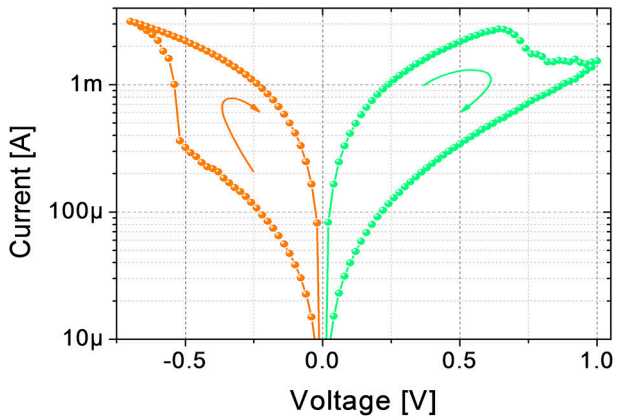


FIGURE 1 | DC characterization of the device. Transition from LRS to HRS is obtained with a DC sweep from 0 V to 1 V, transition from HRS to LRS is obtained with a DC sweep from 0 V to −0.7 V.

and the pulse duration is fixed at 100 μs. Measurements are performed using the custom instrument described in Berdan et al. (2015). In the second experiment, the trains of identical pulses are constituted by 300 repetitions of −550 mV—high and 25 μs—long pulses for potentiation and 300 repetitions of 700 mV—high and 20 μs—long pulses for depression. This second pulse scheme is implemented by a custom setup interfacing High Voltage Semiconductor Pulse Generator Unit (B1525A) and Source Measuring Units of a B1500A. The motivation for the choice of the spike parameters will be given in Section 3. In both experimental procedures, reading operation is carried out using a voltage amplitude which induced no changes in the device resistance.

STDP experiments are carried out placing the device between two spiking channels, i.e., two Waveform Generator/Fast Measurement Units (B1530A) of the already mentioned B1500A, acting as spiking neurons. The relative timing between the two overlapping spikes from the two neurons is mapped in a voltage amplitude, as it will be described in Section 3.2.

The SNN is developed and simulated in MATLAB® environment. The network is a simple fully connected winner-take-all SNN of 30 integrate-and-fire neurons, of which 25 are pre-neurons and 5 post-neurons. The pre-neurons are arranged in a 5 × 5 layer and each pre-neuron is connected to all the post-neurons through 125 artificial synapses. The learning method is unsupervised and the experimental STDP data used to update the synaptic weights during learning are collected in a look up table. The operating principle of the network will be described in detail in Section 3.3. Using the same MATLAB® software, a graphic user interface (GUI) is developed to enhance the software usability (further details in the Supplementary Figure 1).

3. RESULTS

The tests described in the following are carried out in order to provide a thorough overview of the device behavior which is finally exploited in a simple example of neuromorphic computation. The present section is therefore divided in three parts. In the first one, long-term functional plasticity is investigated through two different spiking algorithms, which are exploited to achieve a form of STDP learning rule, in the second part. Finally, a SNN is presented.

3.1. Long-Term Functional Synaptic Plasticity

The plasticity of the device is investigated through two different spiking stimulations, which are fundamental to achieve the shape of STDP required in learning.

Figure 2A shows the evolution of the device resistance during some potentiation and depression cycles (top panel) using trains of spikes of fixed time width and increasing amplitude (bottom panel). The maximum voltages for potentiation (−650 mV) and depression (1.2 V) are those leading to a maximum resistance change of about one order of magnitude and are close to the maximum voltages used in DC operation (**Figure 1**). During depression (resistance increase, green circles), the first spikes, corresponding to lower voltages (see bottom panel of **Figure 2A**), do not induce any resistance change up to a voltage threshold which can be identified at about 550 mV. As the threshold is overcome, the resistance starts increasing gradually. The device therefore presents several intermediate resistive states throughout the programming window. Similarly, during potentiation (resistance decrease, orange circles), several intermediate states are reached between the maximum and minimum resistances using spikes with increasing voltage amplitude. It can be noted that, in this case, the resistance change begins at different voltage levels from cycle to cycle, but for voltages higher than −500 mV a resistance decrease can always be observed. Therefore, −500 mV is considered the voltage threshold for potentiation. It is worth noticing that time widths, as well as voltages, influence the resistance evolution, as already reported by Covi et al. (2015) for similar devices. On the other hand, resistance changes are more sensitive to voltage variations rather than to time widths variations, so that for time widths in the range of 10 to 100 μs roughly the same voltages can be applied for obtaining the same resistance evolution. It has to be mentioned that a stair-case like algorithm, like the one used here, is not practical to implement in real large-scale system, because requires neurons to keep track of previous activity. On the other hand, the testing procedure reported in **Figure 2A** is useful for characterizing the device and to clarify the functioning principle of the STDP implementation described below, which has actually been proposed as a learning rule for practical implementation of neuromorphic hardware (Saïghi et al., 2015).

In the set of measurements shown in **Figure 2B**, plasticity is investigated as a function of trains of identical spikes. Some depression/potentiation cycles are performed. During both potentiation and depression, the resistance gradually changes, featuring several intermediate states between the LRS and the HRS. In all the cycles, the resistance rate change is not constant with respect to the number of spikes. Indeed, for both potentiation and depression the resistance change is faster for the first spikes. In general, analog resistance variation due to trains

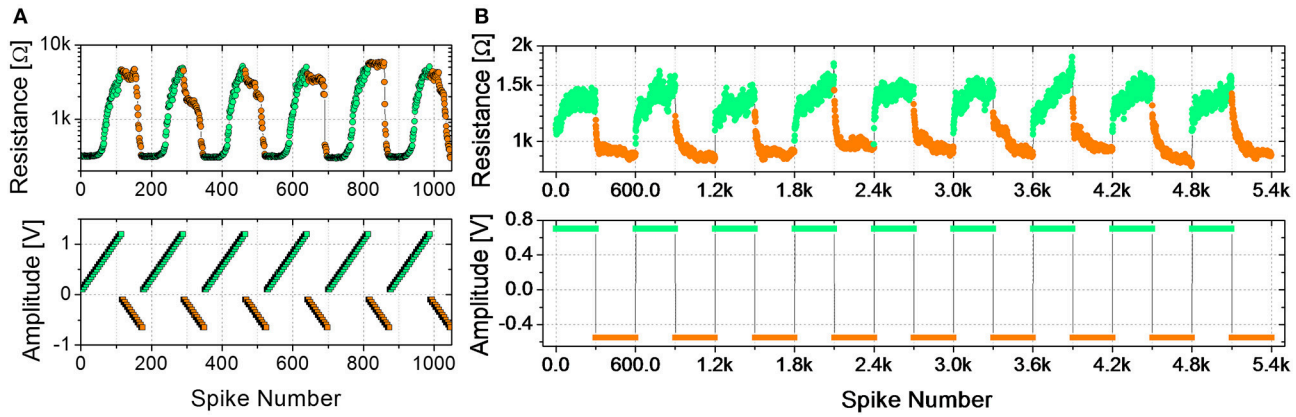


FIGURE 2 | (A) Potentiation (orange) and depression (green) cycles using ramped trains of spikes. Spike time width 100 μs, 5 repetitions. Potentiation: ramps from −0.1 V to −0.65 V. Depression: ramps from 0.1 V to 1.2 V. Upper graph: device resistance after spikes; lower graph: spike amplitude. **(B)** Potentiation (orange) and depression (green) cycles using trains of 300 identical spikes. Potentiation: spike amplitude −0.55 V, time width 25 μs. Depression: spike amplitude 0.7 V, time width 20 μs. Upper graph: device resistance after spikes; lower graph: spike amplitude.

of identical spikes can be found for voltages values close to the voltage thresholds, identified by the results of voltage staircase stimulation for similar time widths (as that shown in **Figure 2A**). Indeed, gradual resistance change is achievable as an intermediate regime between a low voltage stimulation, which does not affect the resistance, and a high voltage stimulation, which induces a digital behavior (Covi et al., 2015). The resistance window obtained through identical pulses is in the order of 2, which has been considered sufficient when dealing with neuromorphic systems (Kuzum et al., 2012; Prezioso et al., 2016).

3.2. Homosynaptic Input-Specific Plasticity Toward Learning

Based on the plasticity results described in Section 3.1 as a function of voltage modulation and spike repetition, STDP experiments relying on engineering of pre- and post-spike superimposition are carried out. Indeed the voltage drop on the memristor is modulated according to the voltage difference resulting from the superimposition of pre- and post-spike waveforms, which depends on their relative timing. To this aim, pre-spike is shaped as a triangular-like pulse (**Figure 3A**), thus acting as a bias performing the voltage-to-time mapping. The rectangular-like shape of the post-spike (**Figure 3A**) determines the supra threshold spike width. **Figure 3B** reports two examples of the superimposition of pre- and post-spikes giving either potentiation or depression and **Figure 3C** reports the quantitative voltage-to-delay-time mapping. In particular, the resulting maximum voltage dropping on the device depends on Δt and varies between −650 mV for potentiation and 800 mV for depression.

To emulate STDP with $\Delta t > 0$ ($\Delta t < 0$), first the device is brought in its HRS (LRS) with a DC sweep, then 250 identical pairs of pre- and post-spikes are applied to the top and bottom electrodes of the device, respectively, keeping Δt constant. The experiment is repeated for different delay times (Δt) and each time the parameter Δt is varied, the device is reinitialized accordingly. **Figures 4A,B** show the device resistance evolution as a function of spike pair repetitions for different delay times in both potentiation (**Figure 4A**) and depression (**Figure 4B**). During potentiation and for every delay time, resistance decreases quickly in the initial phase (about ~25 repetitions) before slowing down markedly in later phases (please notice the vertical scale as going like R_0/R with the increase of the number of spikes, in qualitatively agreement with **Figure 2B**). The same qualitative trend is respected also during depression (**Figure 4B**): the first 10–20 spike pair repetitions significantly change the resistance, whereas the following ones are less effective, until a saturation level is reached after ~150–200 spikes. In both potentiation and depression, the variation of Δt, i.e., the voltage drop, drives the amplitude of the resistance change, i.e., the longer the delay time, the lower the change in resistance. Moreover, Δt affects the resistance change rate in the initial stage of the plasticity operation, i.e., the smaller the delay time (i.e., the higher the voltage drop), the sharper the resistance evolution (e.g., compare the blue and pink curves of **Figures 4A,B**).

Figures 4C,D show the STDP curve represented as the normalized resistance change as a function of the spike delay (and consequently of the voltage amplitude, as shown in the top x-axis of **Figures 4C,D**) for few representative fixed numbers of spike pair repetitions (1, 10, 25, 50, 100, and 150). The plots, which are derived from aforementioned results, qualitatively follow the biological STDP curve shown in Bi and Poo (1998). In accordance with **Figures 4A,C** shows that when Δt is positive and small, the first spike pair induces a resistance variation equal to 75% of the dynamic range. As a consequence, the following repetitions have a reduced effect in further changing the device resistance. On the contrary, when Δt is longer and the resulting spike voltage amplitude is lower, the spike repetitions play an important role in the evolution of the device resistance. Indeed, it becomes progressively more pronounced with increasing Δt. This effect is valid up to a point where Δt is so large that the voltage

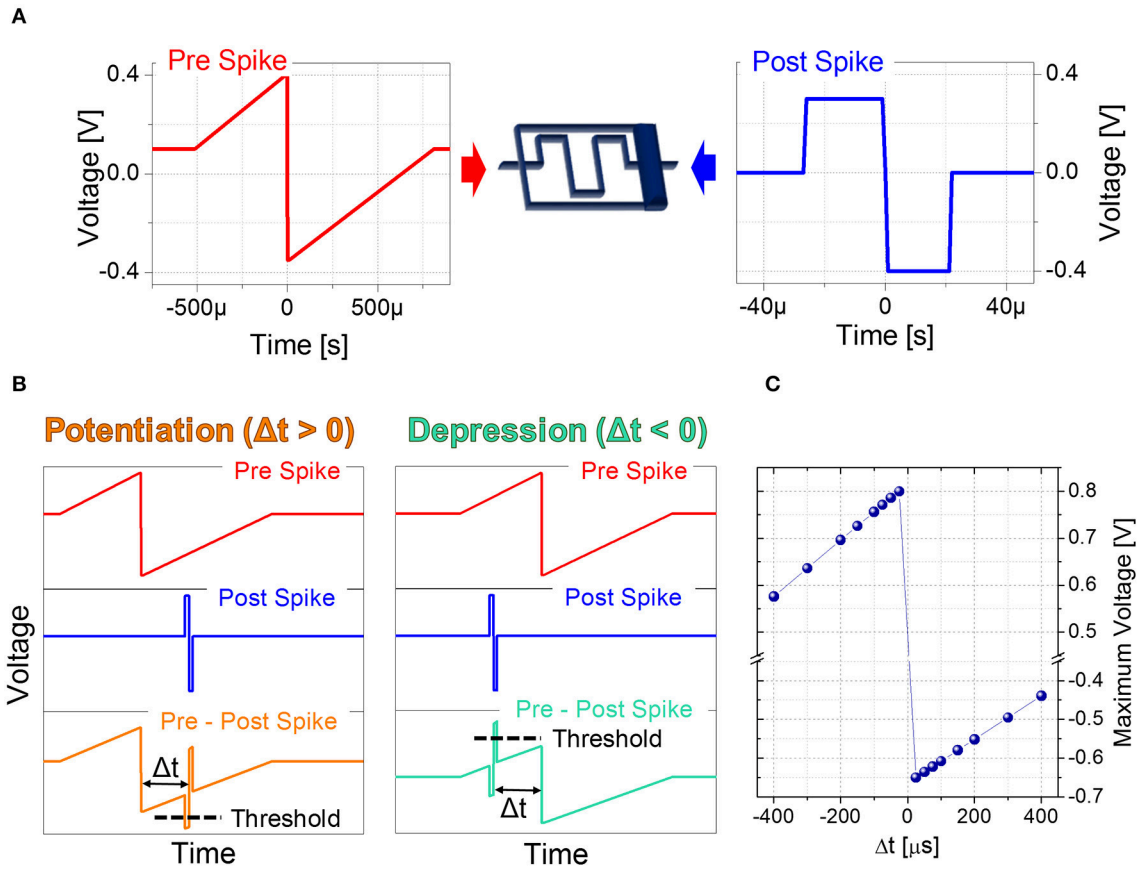


FIGURE 3 | (A) Setup for Spike Time Dependent Plasticity and waveforms used as pre-spike (left) and post-spike (right) in STDP experiments. **(B)** Overlapping of pre-spike and post-spike to obtain a potentiation (left) and a depression (right). **(C)** Voltage-to-delay time mapping. Resulting voltage across the artificial synapse as a function of Δt.

drop across the device does not exceed the device threshold and no more changes in device resistance are induced, regardless of the number of applied spikes. The same effect is shown also in **Figure 4D**, where results for negative Δt are plotted, even though here the effect is less pronounced. Indeed, a change in the synaptic weight is present also for $\Delta t = -400\ \mu s$. This result is in agreement both with **Figure 2A**, where the effect of the voltage amplitude on the device resistance is shown, and with **Figure 2B**, where it is demonstrated that the weight change progressively decreases with increasing spike repetition number.

It is worth mentioning that when the device behavior is tested for $\Delta t > 0$ ($\Delta t < 0$), the device is first brought in its HRS (LRS). In case the memristor in the LRS (HRS) is subjected to pulses with $\Delta t > 0$ ($\Delta t < 0$), no changes in its resistance would occur, since the synapse is already completely potentiated (depressed). This is explicitly shown in **Figures 4C,D** , where for negative (positive) delay times no resistance changes are shown.

From **Figures 4C,D**, a behavioral difference between potentiation and depression dynamics emerges. Despite in both cases the final resistance is strongly influenced by the applied voltage amplitude, during potentiation the applied voltage affects the change in the device resistance starting from the very first spike pair, whereas during depression the effect of the voltage is more evident from the second spike pair on, rather than in the first. Such asymmetry of the curve, even though in principle improvable by optimizing the spike shapes, does not affect the possibility of using the STDP rule for a neuromorphic network.

3.3. Associative Unsupervised Learning in Spiking Neuromorphic Networks

The goal of the following Section is to demonstrate the operation of a small unsupervised network which makes use of the plastic response of the memristor described above to emulate the functionality of a synapse. To this end, we concentrate just on a network with fixed timings, i.e., restricting for simplicity to a subset of the STDP data presented in Section 3.2. More specifically, the curve with $\Delta t = 300\ \mu s$ of **Figure 4A** is selected for potentiation and the one with $\Delta t = -50\ \mu s$ of **Figure 4B** for depression. Of course, the shape of the STDP curve provides additional degrees of freedom that can be exploited for addressing more biologically plausible learning algorithms, e.g., for the treatment of gray-scale or color images. However, such applications go beyond the scope of the present manuscript.

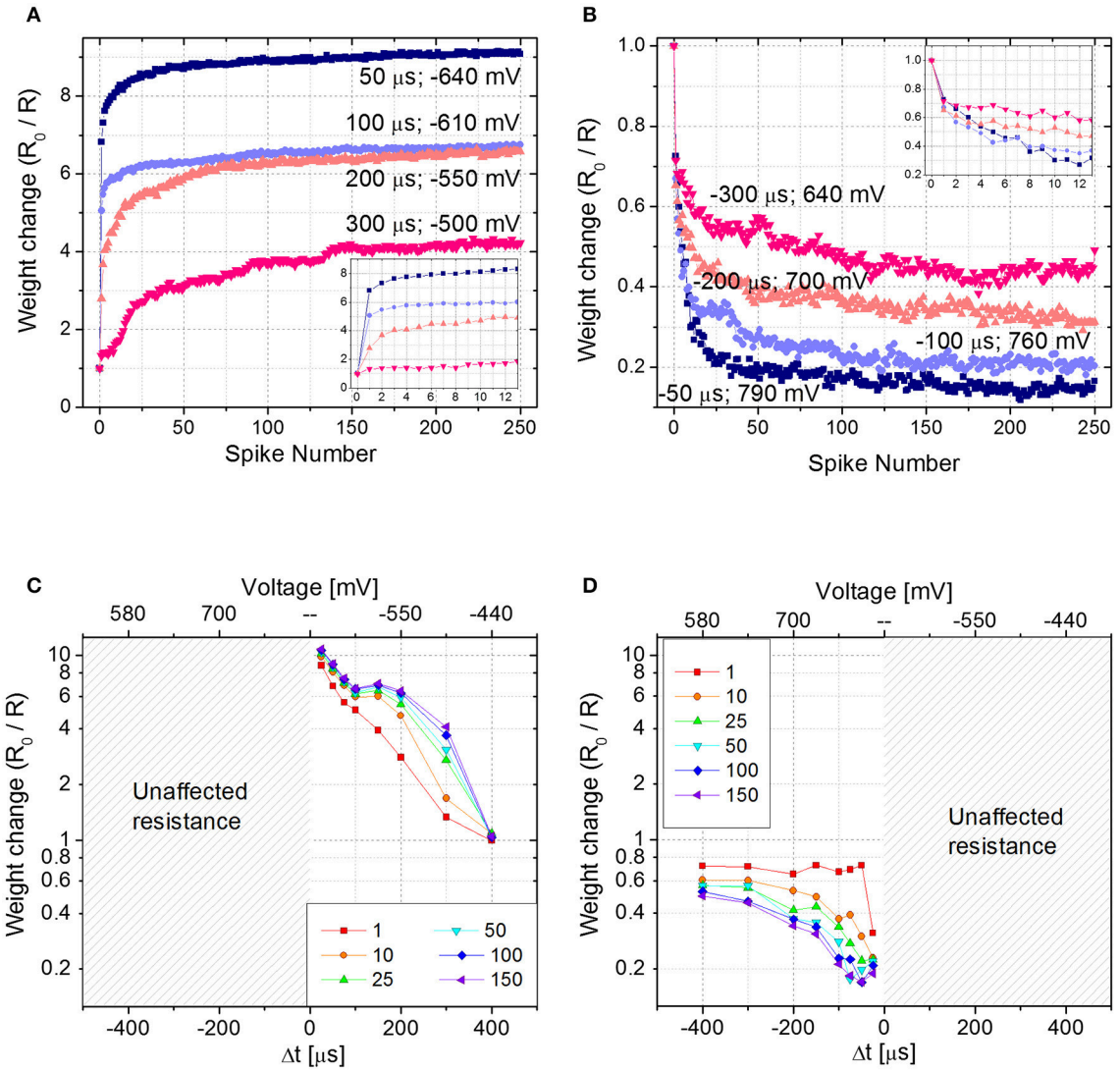


FIGURE 4 | (A) Potentiation and **(B)** depression dynamics with 250 identical spikes. Different voltage amplitudes and delay times are explored. The values of both voltage amplitude and Δt are written nearby each curve. Insets: detail of the first 12 spikes. **(C,D)** Spike Time Dependent Plasticity learning curve for different number of pre- post-spikes pair repetitions ($\Delta t > 0$ and $\Delta t < 0$). R_0 is the initial HRS **(C)** and LRS **(D)**.

Figure 5 shows the proposed SNN. For ease of visualization, in **Figure 5** only a limited number of the connections between pre- and post-neurons is shown. Each of the 25 pixels composing the images is associated to a different pre-neuron. Initially, the network is untrained and a learning phase is executed. At the end, the SNN is able to recognize 5, capital characters (*A*, *E*, *I*, *O*, and *U*, **Figure 5**, inset) given as 5×5 pixel black-and-white images. The network learns through an unsupervised STDP protocol. Once the training session is over, the network is able to recognize incomplete or noisy images, representing any of the characters, following a winner-take-all approach.

The plasticity of the memristor plays most of the role in the learning session of the SNN. The training is performed one character at a time. As an example, the procedure to make the network learn letter *A* is described. The same procedure is then used for all the other characters. The spiking diagram of the neurons is shown in **Figure 6A** and it will be explained together with the unsupervised learning protocol.

At first, character *A* is shown to the network. Black pixels stimulate the associated pre-neurons (**Figure 5**), which fire toward all the post-neurons (**Figure 6A**, top panel). Post-neurons integrate the signals and the one which first reaches its threshold voltage (e.g., post-neuron γ), which is fixed and equal for all the post-neurons, fires back to all the pre-neurons (**Figure 6A**, middle panel). The fired spike has three effects: (i) the discharge of all the other post-neurons, following the winner-take-all rule; (ii) the potentiation of the synapses connecting pre-neurons associated with black pixels and post-neuron γ ($\Delta t > 0$); (iii) the triggering of the firing of the pre-neurons associated with white pixels (**Figure 6A**, bottom panel). Afterward, about 500 μs

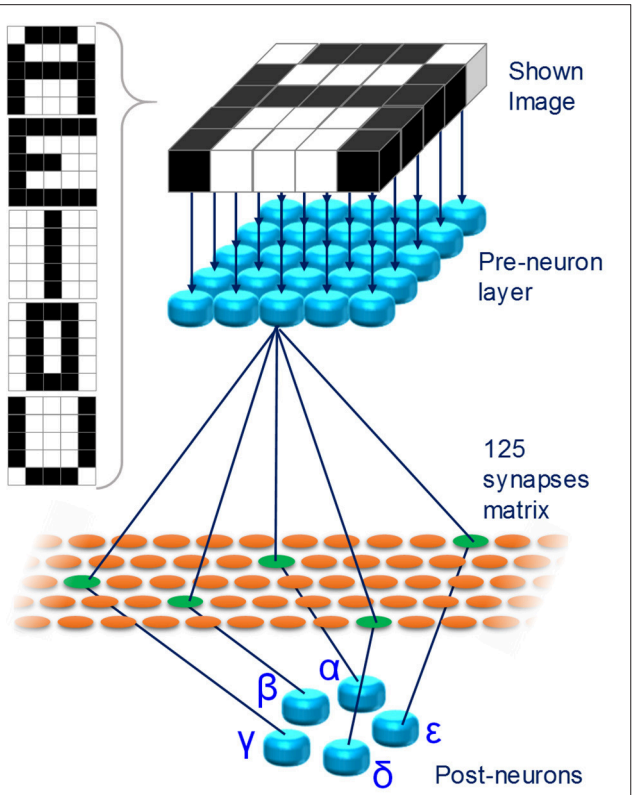


FIGURE 5 | Proposed fully connected SNN. 25 pre-neurons are connected to 5 post-neurons through a layer of 125 artificial synapses. Each pixel of the images shown to the network are associated with a pre-neuron. Inset: images showed to the SNN during the training phase.

after the first spike, post-neuron γ fires again (**Figure 6A**, middle panel), thus depressing the synapses connecting it with firing pre-neurons ($\Delta t < 0$). Pre-neurons associated with black pixels are in their absolute refractory period, therefore the second spike form post-neuron γ has no effect on them. This procedure of neurons handshaking, lasting about 2.15 ms, is called epoch and it occurs each time an image is presented to the SNN during training session. To reach successful learning (i.e., each post-neuron is specialized for a different character) with a probability of 99%, the same character is shown to the network up to 200 times (epochs).

Figure 6B shows an example of training session for letter *A*. The Figure is an excerpt extracted from the video VideoS1.mp4, which can be found in the Supplementary Material and it summarizes the whole 200 epochs occurred to specialize the SNN to recognize character *A*. In panel (i), the image shown to the network is represented. In panel (ii), the synaptic weight after 200 epochs of the subset of synapses contributing to the firing of post-neuron γ is shown. The potentiated synapses are the orange squares in the panel, whereas the depressed ones are colored in black. A close correspondence of panels (i) and (ii) is evident, which is at the basis of the relationship between potentiated synapses and character learned. In panel (iii), the weight evolution as a function of the number of epochs is shown. The depressed synapses (black lines) tend to converge to the lowest conductance value of about 800 μS. On the contrary, the potentiated synapses (orange lines) show a very slight change in the conductance, if any, due to the limit imposed by the initial condition of the synaptic layer. Indeed, the initial conductance of each synapse is set in the range from 1.8 to 2.5 mS. The initial distribution is the result of a potentiation operation and it simulates the device-to-device variability plausible in a real network. Both the width and the average value of the initial weight distribution are fundamental to allow the SNN to uniquely specialize post-neurons during learning session. The variability in the initial resistance, which is actually unavoidable for real devices, allows one post-neuron to be favored with respect to the others and therefore to fire first. The narrower the distribution of initial synaptic weight toward high conductance values, the higher the probability of success during learning. This is true up to the unrealistic situation where all the synapses have the same weight and, therefore, all post-neurons would fire simultaneously, thus failing the learning task. Similarly, the widening of the initial state range leads to a situation where two similar characters, e.g., *E* and *U*, fall in the basin of attraction of the same post-neuron, thus resulting in an unsuccessful learning (i.e., the SNN forgetting the former character and specializing the same post-neuron to recognize the latest character presented). The same erroneous behavior is obtained if the average value of the initial distribution is moved toward lower conductances.

An example of complete training session is illustrated in **Figure 6C** (an animation of the first 50 epochs is shown in Supplementary Material, VideoS2.mp4). Each 5×5 matrix in **Figure 6C** represents the group of 25 synapses contributing to the firing of post-neurons α to ϵ. Initially, all the weights are randomly distributed between 1.8 and 2.5 mS. Increasing the number of epochs (in the Figure, initial state, 5th, 50th, and 200th epochs are shown), the weight of each synapse gradually changes until, at the 200th epoch, the SNN is trained and the characters are recognizable also in the synaptic layer. In addition, **Figure 6D** evidences the distribution of all the 125 synaptic weights in the initial states and after 5, 50, and 200 epochs. It can be noted that during the session the initial distribution, which is initially grouped unimodally toward the highest conductive values, is split in two, one group for depressed synapses and one for potentiated ones, which is consistent with the results shown in **Figure 6B**, panel (iii).

Similar to the training session, during recognition, when an image is shown to the SNN, the stimulated pre-neurons fire toward all the post-neurons. The post-neuron which is first charged above its threshold fires, both recognizing the character shown and discharging the other post-neurons.

The recognition tests are carried out on 100 networks configurations resulting from the same number of learning simulations with different initial synaptic weights. The test set can be divided into two classes of images, one with missing pixels and one with additive noise (Supplementary Figure 2). In the first test, several images with missing black pixels are shown to the SNN. The results demonstrate that in the worst case the network is always (100% recognition rate) able to recognize the character if the percentage of missing pixel is equal to or lower than 21% for character *A*, 27% for character *E*, 20% for character *I*, 33% for

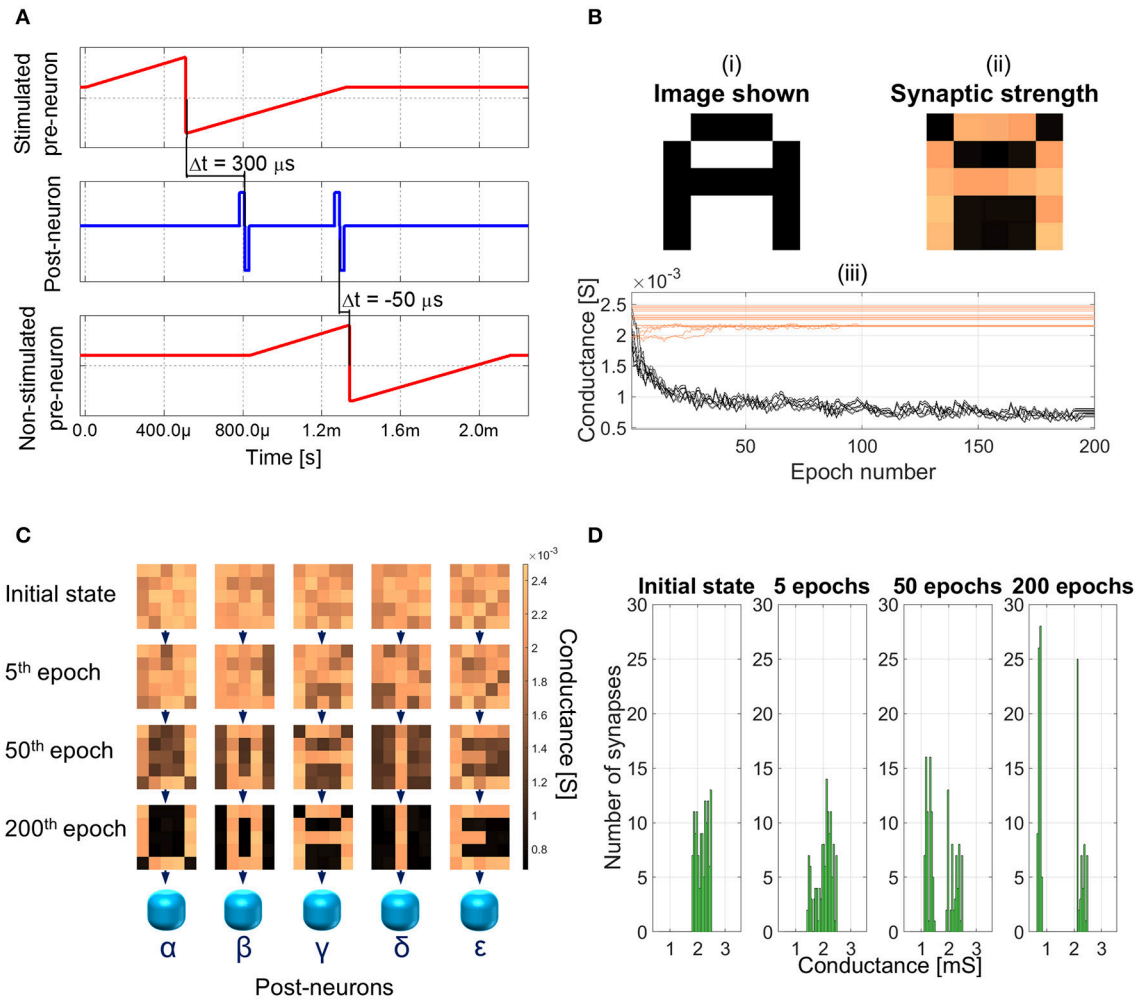


FIGURE 6 | (A) Training session: spiking diagram of one epoch. The training character is shown at 0 s and the duration of an epoch is about 2.15 ms. **(B)** Image shown to the network (top left panel), synaptic weights after 200 epochs (top right panel), and detailed synaptic weight evolution during training session of character *A* (bottom panel). Black lines represent the synapses which are being depressed during the session and orange lines the ones potentiated. **(C)** Example of synaptic weight changes during a learning session. Each 5 × 5 matrix represents the group of 25 synapses contributing to the firing of neurons α to ϵ. Color bar on the right indicates the conductance range of the synapses. Increasing the number of epochs (from top to bottom), the SNN specializes each post-neuron to recognize a different character. **(D)** Distribution of the synaptic weights during the training session.

character *O*, and 18% for character *U*. In the second test, noisy images are shown to the network. The test images are chosen among the ones considered mostly critical for the SNN to be recognized, so that worst cases could be explored. Further details about the images shown and the choice criterion can be found in the Supplementary Figures 2, 3, and Supplementary Table 1. The network recognition rate resulted 85.71% for images with up to 4 noise pixels. However, the recognition rate is correlated with the number of epochs in the training session. As already mentioned, a training session for a character consists of 200 epochs and it almost always leads to a successful learning. If the number of epochs during training is reduced, both the success rate of the learning session and the recognition rate decrease. Simulations of learning sessions with different number of epochs (200, 50, 10, 8, and 5) are carried out. With a number of epochs of 8, 2 learning sessions out of 3 failed, and with 5 epochs the SNN can never perform a successful learning. After concluding a successful learning session, the same test images (see Supplementary Figure 2) are shown to the SNN during recognition. The recognition rate decreases from 88.22% (200 epochs) to 82.61% (50 epochs), 75.29% (10 epochs), and 72.03% (8 epochs). This means that, when a limited number of epochs is performed, the synapses may be insufficiently depressed and during recognition they may conflict with the potentiated ones, thus resulting in incorrect recognition.

4. DISCUSSION

In Section 3 a filamentary HfO_2 memristor featuring analog behavior is presented. The proposed device is able to emulate

both long-term plasticity and STDP learning rule. Moreover, a simple fully-connected SNN which takes advantage of the memristor plastic behavior and which uses an associative unsupervised STDP-based learning protocol is simulated. After a training session the network is able to recognize five characters, even when the images displayed are incomplete or noisy. It should be mentioned that non-ideal elements, such as parasitic or jitter, are deliberately not considered in the proposed network, because the performed investigation focuses on the basic principles of the network with analog memristor, where a study at a high level of abstraction is mandatory before considering practical implementations.

We demonstrate long-term functional plasticity with two different spiking algorithms, which have been already used in the literature (Park et al., 2013; Yu et al., 2013a; Li et al., 2014; Zhao et al., 2014) to emulate plasticity. The two algorithms allow an investigation on the device behavior as a function of the voltage amplitude (**Figure 2A**) and on its integrative response when stimulated by identical spikes (**Figure 2B**). An algorithm that modulates the spike voltage applied to the device is not easy to be implemented in a system. Indeed, dedicated read-out and variable voltage biasing circuits are required. On the other hand, the voltage on the memristor in a system could be modulated through superimposition of long spikes, as proposed several times in literature (Serrano-Gotarredona et al., 2013; Saïghi et al., 2015). This method allows the neuron to always fire the same spike and let the delay times between spike determine the actual voltage on the device.

The combined results of the measurements shown in **Figure 2** are used to engineer the shape of pre- and post-spikes used to emulate homosynaptic plasticity and to conceive a biologically plausible STDP curve (**Figures 4C,D**), which takes advantage of both the relative timing between the two spikes (Δt) and the plasticity given by spike pair repetition. It should be noted that, though analog changes can be obtained around the previously found thresholds for potentiation and depression, the device can be operated in an analog fashion in a range of voltage of some hundreds of mV (**Figure 4**). From **Figures 4A,B**, it can be observed that voltages from 580 to 800 mV for depression and from −440 to −650 mV for potentiation allow a resistance evolution as a function of the repetition of identical spikes. In particular, **Figures 4C,D** show that the dynamic range decreases with the decreasing of the applied voltage, but resistance still gradually changes. In a network, it can be expected that different devices show analog transitions for a range of voltages whose end values (V_{min}, V_{max}) can be different from device to device, but in general a sub-range of voltages allowing analog resistance modulation is shared by many devices. A threshold difference in the devices (provided it is within few tens to one hundred mV) would not prevent analog behavior, as demonstrated in **Figure 7**, which shows the behavior of 3 different devices during potentiation (**Figure 7A**) and depression (**Figure 7B**) when stimulated by trains of 300 identical spikes. In both **Figures 7A,B**, the mean value of 10 repetitions of the same train of spikes is represented by symbols and the shaded area indicates the standard deviation of the measurements. It can be noted that potentiation suffers of major variability with respect to depression. Nevertheless, despite the device-to-device variability, all the devices show an analog behavior in both operations. In addition, different resistance evolutions due to different device thresholds are compensated in SNNs by the high parallelism of the architecture itself which enhances the network tolerance to device variability (Yu et al., 2013a). In this respect, the performance of the presented SNN against variability is tested adding ±10% (**Figures 8A,B**) and ±30% (**Figures 8C,D**) device-to-device variability in the artificial synapses behavior, i.e., the look up table associated to each synapse has been multiplied by a random factor extracted between 0.9/1.1 and 0.7/1.3 respectively. **Figure 8A** summarizes the synaptic weight evolution during the training session of all the characters as a function of the epoch number when a variability of ±10% is set. Each graph shows the weight evolution of the group of synapses contributing to the firing of a specific post-neuron. During learning, depression (black) and potentiation (orange)

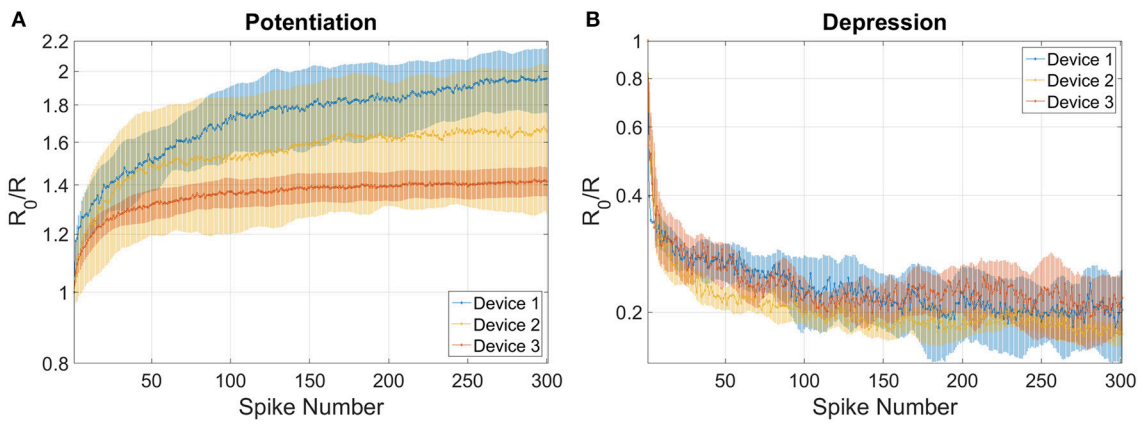


FIGURE 7 | Variability in the behavior of 3 different devices for (A) potentiation and (B) depression when stimulated by trains of 300 identical spikes. Potentiation: voltage amplitude −0.55 V, time width 25 μs. Depression: voltage amplitude 0.75 V, time width 20 μs. Symbols indicate the mean value of 10 repetitions of the same train of spikes and the shaded area indicates the standard deviation.

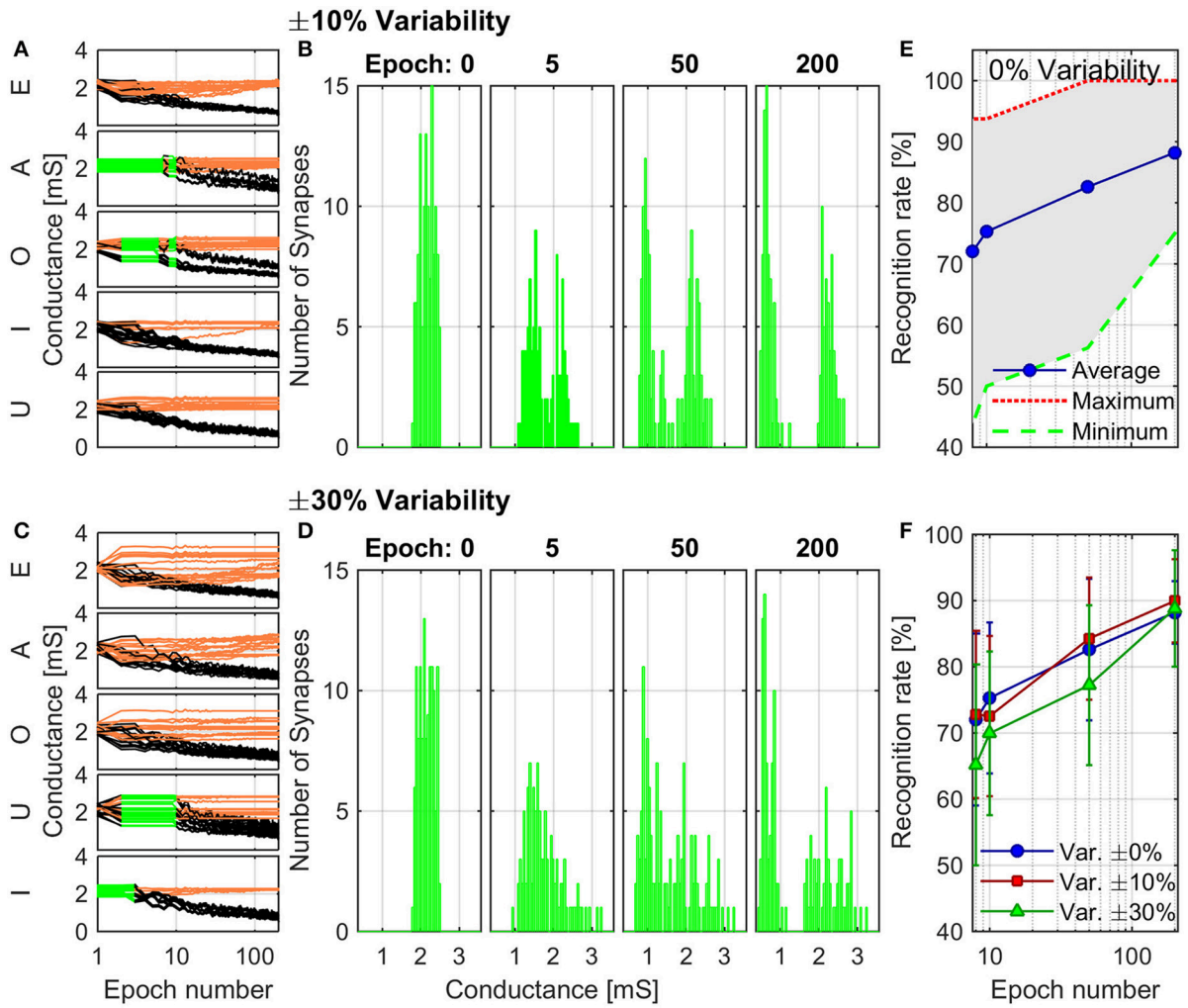


FIGURE 8 | Simulation of the training session including ±10% (A,B) and ±30% (C,D) of variability in synaptic behavior. (A,C) Detailed synaptic weight evolution during training session of all characters. Black lines represent the synapses which are being depressed during the session and orange lines the ones potentiated. Green lines indicate that the neuron did not fire in the corresponding epoch. **(B,D)** Distribution of the synaptic weights during the training session. **(E)** Recognition rate as a function of the number of epochs in the learning session. The blue circles represent the average recognition rate from 100 simulations where device-to-device variability is not taken into account. The red dotted line and the green dashed one indicate the best and worst results obtained in the simulations, respectively, whereas the other results lie in the shaded gray area. **(F)** Average recognition rate of 100 simulations as a function of the number of epochs in the learning session with device-to-device variability of 0% (blue circles), ±10% (red squares), and ±30% (green triangles). Error bars show the standard deviation of the results.

of synapses occur, but the weight evolution with and without variability (as in **Figure 6B**) is different, because in the former case for some presentation of the images to the network some groups of synapses are not updated (green lines for synapses connecting to post-neuron that is finally specialized to characters *A* and *O*). This is explained as follows. In the examples reported in **Figure 8**, first, *O* is presented and post-neuron *O* (meaning post-neuron that finally specializes to recognize *O*) starts firing and updating its associated synapses in the first epoch. On the other hand, variability causes that the weight are adjusted in such a way that from epoch 2 to 6, a different post-neuron fires and synaptic weights associated to post-neuron *O* are frozen. Then, specialization proceeds with one post-neuron specializing for only one character. The success of the learning session demonstrates the robustness of the network against device-to-device variability, in accordance with Yu et al. (2015), provided analog behavior holds in each device. **Figure 8B** shows the weight distribution of the synaptic matrix during training. Increasing the number of epochs, the initial synaptic weight distribution tends to separate in two groups, one for depressed synapses and one for potentiated synapses, as it happens also in **Figure 6D**. However, in the case of **Figure 8B**, the two distribution are wider than in the case where no variability factor is considered. The same above observations are valid also when variability is increased to ±30%, as shown in **Figures 8C,D**. Indeed, also **Figure 8C** shows, in the bottom two graphs, some epochs where the synaptic weight is not updated. Moreover, considering the ±30% variability test, the final distribution of the synaptic

weights is larger than the one achieved for ±10% variability (17% larger for depression and 136% larger for potentiation). In this respect, it is worth analyzing the recognition rate of the test set shown in Supplementary Figure 2, as a function of the number of epochs carried out during learning. **Figure 8E** shows the recognition rate (blue circles) as a function of the number of epochs in a SNN neglecting device variability. Each circle is the average recognition rate over 100 simulations (i.e., 100 learning sessions each starting with a different initial configuration of the synaptic weights) and the results of each simulation lie in the gray shaded area delimited by the best simulation result (dotted red line) and the worst one (dashed green line). The increase of the number of epochs during learning improves the average recognition rate and decreases the spread of the results. Indeed, the recognition rate varies between 43.75 and 93.75% at 8 learning epochs whereas it varies between 75 and 100% at 200 learning epochs. As already mentioned in Section 3.3, the recognition rate is closely related to the distribution of the synaptic weights at the end of the training session. The nearer the distributions of the potentiated and depressed synapses, the lower the recognition rate. As a consequence, the increase of the number of learning epochs contributes to enhance the separation of the two above-mentioned distributions and, therefore, to improve the recognition rate. It is interesting to note that in this respect the impact of device-to-device variability is almost negligible. Indeed, we performed the same recognition tests with the same methodology also in case of SNNs with ±10 and ±30% device-to-device variability. **Figure 8F** shows the average recognition rate as a function of the epochs during learning in case of 0% (blue circles), ±10% (red squares), and ±30% (green triangles) device-to-device variability. The vertical bars indicate the standard deviation σ. The same increasing trend can be noted for all the curves regardless of the variability. In accordance with **Figure 8E**, in each curve also σ decreases with increasing number of epochs, but the value of σ for each number of epochs during learning increases with increasing variability. On the other hand, the network proves to be robust also for variability up to ±30%. The network robustness lies in the gradual synaptic weight update. Indeed for every post-neuron spike, the weight is adjusted by a small amount. If an erroneous spiking (like the one of a post-neuron responding to two different characters) occurs, the weight change is small enough that the following epochs can recover the error.

Given the observations above, we would like to stress that it is fundamental in deterministic networks to have analog synapses even though, as in the proposed SNN, the images shown are only black and white. Indeed, in a system with deterministic neurons, as in the proposed one, binary deterministic memristors would lead to fast learning (only few epochs would be necessary to complete the training session), but also to fast forgetting (Fusi and Abbott, 2007). Indeed, if a noisy image were shown to a trained SNN employing binary synapses, the network would classify that image and, therefore, would adjust the synaptic matrix also according to the pixel which is not representative for that image, disrupting learning. In the case of analog synapses, the same permanent and significant change leading to failure would result only if the same noisy image were shown to the network for several epochs, which is statistically improbable.

In the presented SNN, using two fixed delay times (one for potentiation and one for depression) in the STDP is sufficient as a proof-of-concept. In this respect, two values are selected ($\Delta t = 300\ \mu s$ and $\Delta t = -50\ \mu s$) which are coherent with a post-neuron firing as a consequence of the stimulation by the pre-neuron (synapses potentiation for $\Delta t = 300\ \mu s$) rather than with a pre-neuron firing because of the stimulation by the activated post-neurons in case of synaptic depression ($\Delta t = -50\ \mu s$). On the other hand, a network exploiting also the possibility of variable delay times between pre- and post-spikes, would allow increasing the available resistance states, therefore, improving the network robustness even further. As an example, in the case of input-specific associative learning rules for pattern recognition, the possibility to combine different parameters (Δt and spike pair repetition) to achieve various resistive states with different evolution histories offers a further degree of freedom. Indeed, a possible application could be in networks where images have different colors o shades of gray, which can be linked to different delay times. In this case, at the end of a learning session with a certain number of epochs, the weight distribution of the synaptic matrix would give an indication of the common features of the various images presented to the network. More specifically, the more a group of synapses is potentiated, the more they are stimulated, i.e., the potentiated group identifies a common feature in the set of displayed images.

5. CONCLUSION

In summary, a thorough analysis of the synaptic features of the proposed oxide-based memristor is carried out. Initially, the device ability to emulate long-term functional potentiation and depression is proved upon stimulation with spikes with increasing amplitude (stair-case like) and trains of identical spikes. These experiments show that the memristor has an analog behavior in tuning its resistance and it can reach a dynamic range up to one order of magnitude depending on the spiking algorithm employed. Then, homosynaptic plasticity is tested through STDP experiments, which demonstrates the device biological-like behavior when subjected to synaptic activity. Finally, the possibility of developing deterministic networks using unsupervised learning is investigated. A subset of the STDP collected data is used to simulate a simple fully-connected SNN featuring an associative unsupervised STDP-based learning protocol. The network is able, after a training session, to recognize the five characters, also when partially incomplete or noisy letters are displayed. Therefore, the SNN proves that the proposed memristor can be used to emulate the functionality of an artificial synapse in future neuromorphic architectures with deterministic neurons, and analog memristive synapses, and making use of unsupervised learning for real-time applications.

AUTHOR CONTRIBUTIONS

EC, SB, and SS conceived the experiments and wrote the manuscript. SB and SS developed the memristor device. EC

and SB collected the data on synaptic plasticity, in collaboration with AS. EC and SB performed the STDP experiments. EC and SB developed the SNN, in collaboration with AS. All authors discussed the results and contributed to manuscript preparation.

FUNDING

The work has been partially supported by the FP7 European project RAMP (grant agreement n. 612058).

ACKNOWLEDGMENTS

The authors acknowledge Dr. M. Alia for his support in device fabrication.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnins.2016.00482

REFERENCES

Ambrogio, S., Balatti, S., Milo, V., Carboni, R., Wang, Z. Q., Calderoni, A., et al. (2016a). Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM. *IEEE Trans. Electr. Dev.* 63, 1508–1515. doi: 10.1109/TED.2016.2526647

Ambrogio, S., Balatti, S., Nardi, F., Facchinetti, S., and Ielmini, D. (2013). Spike-timing dependent plasticity in a transistor-selected resistive switching memory. *Nanotechnology* 24:384012. doi: 10.1088/0957-4484/24/38/384012

Ambrogio, S., Ciocchini, N., Laudato, M., Milo, V., Pirovano, A., Fantini, P., et al. (2016b). Unsupervised learning by spike timing dependent plasticity in phase change memory (PCM) synapses. *Front. Neurosci.* 10:56. doi: 10.3389/fnins.2016.00056

Berdan, R., Serb, A., Khiat, A., Regoutz, A., Papavassiliou, C., and Prodromakis, T. (2015). A μ-controller-based system for interfacing selector-less RRAM crossbar arrays. *IEEE Trans. Electr. Dev.* 62, 2190–2196. doi: 10.1109/TED.2015.2433676

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

Bill, J., and Legenstein, R. (2014). A compound memristive synapse model for statistical learning through stdp in spiking neural networks. *Front. Neurosci.* 8:412. doi: 10.3389/fnins.2014.00412

Brivio, S., Covi, E., Serb, A., Prodromakis, T., Fanciulli, M., and Spiga, S. (2016). Experimental study of gradual/abrupt dynamics of HfO_2-based memristive devices. *Appl. Phys. Lett.* 109, 133504. doi: 10.1063/1.4963675

Brivio, S., Frascaroli, J., and Spiga, S. (2015). Role of metal-oxide interfaces in the multiple resistance switching regimes of Pt/HfO_2/TiN devices. *Appl. Phys. Lett.* 107, 023504. doi: 10.1063/1.4926340

Brivio, S., Tallarida, G., Cianci, E., and Spiga, S. (2014). Formation and disruption of conductive filaments in a HfO_2/TiN structure. *Nanotechnology* 25:385705. doi: 10.1088/0957-4484/25/38/385705

Burr, G. W., Shelby, R. M., Sidler, S., di Nolfo, C., Jang, J., Boybat, I., et al. (2015). Experimental demonstration and tolerancing of a large-scale neural network (165 000 Synapses) using phase-change memory as the synaptic weight element. *IEEE Trans. Electr. Dev.* 62, 3498–3507. doi: 10.1109/TED.2015.2439635

Covi, E., Brivio, S., Fanciulli, M., and Spiga, S. (2015). Synaptic potentiation and depression in Al:HfO_2-based memristor. *Microelectron. Eng.* 147, 41–44. doi: 10.1016/j.mee.2015.04.052

Covi, E., Brivio, S., Serb, A., Prodromakis, T., Fanciulli, M., and Spiga, S. (2016). "HfO_2-based memristors for neuromorphic applications," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)* (Montreal, QC), 393–396.

Du, N., Kiani, M., Mayr, C. G., You, T., Bürger, D., Skorupa, I., et al. (2015). Single pairing spike-timing dependent plasticity in $BiFeO_3$ memristors with a time window of 25ms to 125μs. *Front. Neurosci.* 9:227. doi: 10.3389/fnins.2015.00227

Eryilmaz, S. B., Kuzum, D., Jeyasingh, R., Kim, S., BrightSky, M., Lam, C., et al. (2014). Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array. *Front. Neurosci.* 8:205. doi: 10.3389/fnins.2014.00205

Frascaroli, J., Brivio, S., Ferrarese Lupi, F., Seguini, G., Boarino, L., Perego, M., et al. (2015). Resistive switching in high-density nanodevices fabricated by block copolymer self-assembly. *ACS Nano* 9, 2518–2529. doi: 10.1021/nn505131b

Fusi, S., and Abbott, L. F. (2007). Limits on the memory storage capacity of bounded synapses. *Nat. Neurosci.* 10, 485–493. doi: 10.1038/nn1859

Garbin, D., Vianello, E., Bichler, O., Rafhay, Q., Gamrat, C., Ghibaudo, G., et al. (2015). HfO_2-based oxRAM devices as synapses for convolutional neural networks. *IEEE Trans. Electr. Dev.* 62, 2494–2501. doi: 10.1109/TED.2015.2440102

Kuzum, D., Jeyasingh, R. G. D., Lee, B., and Wong, H.-S. P. (2012). Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Lett.* 12, 2179–2186. doi: 10.1021/nl201040y

Kuzum, D., Yu, S., and Wong, H.-S. P. (2013). Synaptic electronics: materials, devices and applications. *Nanotechnology* 24:382001. doi: 10.1088/0957-4484/24/38/382001

Li, Y., Zhong, Y., Zhang, J., Xu, L., Wang, Q., Sun, H., et al. (2014). Activity-dependent synaptic plasticity of a chalcogenide electronic synapse for neuromorphic systems. *Sci. Rep.* 4, 1–7. doi: 10.1038/srep04906

Mandal, S., El-Amin, A., Alexander, K., Rajendran, B., and Jha, R. (2014). Novel synaptic memory device for neuromorphic computing. *Sci. Rep.* 4:5333. doi: 10.1038/srep05333

Matveyev, Y., Egorov, K., Markeev, A., and Zenkevich, A. (2015). Resistive switching and synaptic properties of fully atomic layer deposition grown TiN/HfO_2/TiN devices. *J. Appl. Phys.* 117, 044901. doi: 10.1063/1.4905792

Nishitani, Y., Kaneko, Y., and Ueda, M. (2015). Supervised learning using spike-timing-dependent plasticity of memristive synapses. *IEEE Trans. Neural Netw. Learn. Syst.* 26, 2999–3008. doi: 10.1109/TNNLS.2015.2399491

Park, S., Chu, M., Kim, J., Noh, J., Jeon, M., Hun Lee, B., et al. (2015). Electronic system with memristive synapses for pattern recognition. *Sci. Rep.* 5, 1–9. doi: 10.1038/srep10123

Park, S., Sheri, A., Kim, J. H., Noh, J., Jang, J., Jeon, M. G., et al. (2013). "Neuromorphic speech systems using advanced ReRAM-based synapse," in *Proceedings of IEEE International Electron Devices Meeting (IEDM)* (Washington, DC).

Prezioso, M., Merrikh-Bayat, F., Hoskins, B., Likharev, K., and Strukov, D. (2016). Self-adaptive spike-time-dependent plasticity of metal-oxide memristors. *Sci. Rep.* 6:21331. doi: 10.1038/srep21331

Prezioso, M., Merrikh-Bayat, F., Hoskins, B. D., Adam, G. C., Likharev, K. K., and Strukov, D. B. (2015). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nat. Lett.* 521, 61–64. doi: 10.1038/nature14441

Querlioz, D., Bichler, O., Dollfus, P., and Gamrat, C. (2013). Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Trans. Nanotechnol.* 12, 288–295. doi: 10.1109/TNANO.2013.2250995

Querlioz, D., Bichler, O., Vincent, A. F., and Gamrat, C. (2015). Bioinspired programming of memory devices for implementing an inference engine. *Proc. IEEE* 103, 1398–1416. doi: 10.1109/JPROC.2015.2437616

Saïghi, S., Mayr, C. G., Serrano-Gotarredona, T., Schmidt, H., Lecerf, G., Tomas, J., et al. (2015). Plasticity in memristive devices for spiking neural networks. *Front. Neurosci.* 9:51. doi: 10.3389/fnins.2015.00051

Serb, A., Bill, J., Khiat, A., Berdan, R., Legenstein, R., and Prodromakis, T. (2016). Unsupervised learning in probabilistic neural networks with

multi-state metal-oxide memristive synapses. *Nat. Commun.* 7:12611. doi: 10.1038/ncomms12611

Serrano-Gotarredona, T., Masquelier, T., Prodromakis, T., Indiveri, G., and Linares-Barranco, B. (2013). STDP and STDP Variations with Memristors. *Front. Neurosci.* 7:2. doi: 10.3389/fnins.2013.00002

Suri, M., Querlioz, D., Bichler, O., Palma, G., Vianello, E., Vuillaume, D., et al. (2013). Bio-inspired stochastic computing using binary CBRAM synapses. *IEEE Trans. Electr. Dev.* 60, 2402–2409. doi: 10.1109/TED.2013.2263000

Wang, Z., Ambrogio, S., Balatti, S., and Ielmini, D. (2015). A 2-transistor/1-resistor artificial synapse capable of communication and stochastic learning for neuromorphic systems. *Front. Neurosci.* 8:438. doi: 10.3389/fnins.2014.00438

Yu, S., Chen, P. Y., Cao, Y., Xia, L., Wang, Y., and Wu, H. (2015). "Scaling-up resistive synaptic arrays for neuro-inspired architecture: challenges and prospect," in *2015 IEEE International Electron Devices Meeting (IEDM)* (Washington, DC). doi: 10.1109/IEDM.2015.7409718

Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., and Wong, H.-S. P. (2013a). A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation. *Adv. Materials* 25, 1774–1779. doi: 10.1002/adma.201203680

Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., and Wong, H.-S. P. (2013b). Stochastic learning in oxide binary synaptic device for neuromorphic computing. *Front. Neurosci.* 7:186. doi: 10.3389/fnins.2013.00186

Yu, S., Wu, Y., Jeyasingh, R., Kuzum, D., and Wong, H. S. P. (2011). An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. *IEEE Trans. Electr. Dev.* 58, 2729–2737. doi: 10.1109/TED.2011.2147791

Zhao, L., Chen, H.-Y., Wu, S.-C., Jiang, Z., Yu, S., Hou, T.-H., et al. (2014). Multi-level control of conductive nano-filament evolution in HfO_2 ReRAM by pulse-train operations. *Nanoscale* 6, 5698–5702. doi: 10.1039/c4nr00500g

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer SJ and handling Editor declared their shared affiliation, and the handling Editor states that the process nevertheless met the standards of a fair and objective review.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 08 March 2016
doi: 10.3389/fnins.2016.00056

Unsupervised Learning by Spike Timing Dependent Plasticity in Phase Change Memory (PCM) Synapses

*Stefano Ambrogio[1], Nicola Ciocchini[1], Mario Laudato[1], Valerio Milo[1], Agostino Pirovano[2], Paolo Fantini[2] and Daniele Ielmini[1]**

[1] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano and IU.NET, Milano, Italy,* [2] *Research and Development Process, Micron Semiconductor Italia, Vimercate, Italy*

We present a novel one-transistor/one-resistor (1T1R) synapse for neuromorphic networks, based on phase change memory (PCM) technology. The synapse is capable of spike-timing dependent plasticity (STDP), where gradual potentiation relies on set transition, namely crystallization, in the PCM, while depression is achieved via reset or amorphization of a chalcogenide active volume. STDP characteristics are demonstrated by experiments under variable initial conditions and number of pulses. Finally, we support the applicability of the 1T1R synapse for learning and recognition of visual patterns by simulations of fully connected neuromorphic networks with 2 or 3 layers with high recognition efficiency. The proposed scheme provides a feasible low-power solution for on-line unsupervised machine learning in smart reconfigurable sensors.

Keywords: neuromorphic circuits, spike timing dependent plasticity, phase change memory, neural network, memristor, pattern recognition, cognitive computing

OPEN ACCESS

Edited by:
Themis Prodromakis, University of Southampton, UK

Reviewed by:
Damien Querlioz, CNRS, University of Paris-Sud, France
Mostafa Rahimi Azghadi, The University of Sydney, Australia
Erika Covi, Institute for Microelectronics and Microsystems, CNR, Italy

****Correspondence:***
Daniele Ielmini
daniele.ielmini@polimi.it

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *30 October 2015*
Accepted: *08 February 2016*
Published: *08 March 2016*

Citation:
Ambrogio S, Ciocchini N, Laudato M, Milo V, Pirovano A, Fantini P and Ielmini D (2016) Unsupervised Learning by Spike Timing Dependent Plasticity in Phase Change Memory (PCM) Synapses. Front. Neurosci. 10:56. doi: 10.3389/fnins.2016.00056

INTRODUCTION

Neuromorphic engineering represents one of the most promising fields for developing new computing paradigms complementing or even replacing current Von Neumann architecture (Indiveri and Liu, 2015). Tasks such as learning and recognition of visual and auditory patterns are naturally achieved in the human brain, whereas they require a comparably long time and excessive power consumption in a digital central processor unit (CPU). To address the learning task, one approach is to manipulate the synaptic weights in a multilayer neuron architecture called perceptron, where neurons consist of CMOS analog circuits to perform spike integration and firing, while synapses serve as interneuron connections with reconfigurable weights (Suri et al., 2011; Kuzum et al., 2012; Indiveri et al., 2013; Wang et al., 2015). Recent advances in nanotechnology have provided neuromorphic engineers with new devices which allow for synaptic plasticity, such as resistive switching memory (RRAM; Waser and Aono, 2007; Jo et al., 2010; Ohno et al., 2011; Ambrogio et al., 2013; Prezioso et al., 2015), spin-transfer-torque memory (STT-RAM; Locatelli et al., 2014; Thomas et al., 2015; Vincent et al., 2015), or phase change memory (PCM; Suri et al., 2011; Bichler et al., 2012; Burr et al., 2014; Eryilmaz et al., 2014). In particular, recent works have shown the ability to train real networks for pattern learning, adopting backpropagation (Burr et al., 2014) and recurrently-connected network (Eryilmaz et al., 2014). The advantage of these devices over CMOS is the small area, enabling the high synaptic density which is required to achieve the large connectivity (i.e., ratio between synapses and neurons) and highly parallelized architecture of the human brain. In addition, nanoelectronic synapses allow for low-voltage operation in hybrid CMOS-memristive circuits, and for augmented functionality with respect to CMOS technology, thanks to the peculiar phenomena taking place in the memristive element.

For instance, the CMOS-memristive synapse showed the ability to perform spike-timing dependent plasticity (STDP; Yu et al., 2011; Ambrogio et al., 2013), the transition from short-term to long-term learning (Ohno et al., 2011), a multilevel cell operation allowing for gradual weight update (Wang et al., 2015) and a stochastic operation suitable to redundant neuromorphic networks (Suri et al., 2012; Yu et al., 2013; Garbin et al., 2015; Querlioz et al., 2015).

In this context, PCM technology is an attractive solution for nanoelectronic synapse in high density neuromorphic systems. PCM is currently under consideration for stand-alone (Servalli, 2009) and embedded memories (Annunziata et al., 2009; Zuliani et al., 2013). Generally, the device appears with one-transistor/one-resistor (1T1R) architecture which allows for strong immunity to voltage variations as well as relatively compact structure. Either metal-oxide-semiconductor (MOS) or bipolar junction transistor (BJT) have been used in the 1T1R architecture. In some case, the one-diode/one-resistor (1D1R) structure has been demonstrated, capable of extremely small area and high density using the crosspoint architecture (Kau et al., 2009). The PCM technology platform has been used for computing applications for Boolean logic functions (Cassinerio et al., 2013) and arithmetic computation (Wright et al., 2011), including numerical addition, subtraction and factorization (Hosseini et al., 2015). Neuromorphic synapses have also been studied: Kuzum et al., have first demonstrated STDP in PCM by use of an ad-hoc train of pulses at either terminal of the device (Kuzum et al., 2012). Suri et al., have presented a 2-PCM synapse, where the 2 PCM devices serve as complementary potentiation and depression via gradual crystallization (Suri et al., 2011; Bichler et al., 2012). Supervised training and learning using back-propagation schemes were recently shown using PCM arrays (Burr et al., 2014; Eryilmaz et al., 2014). Despite the wealth of novel demonstrations of PCM technology, no STDP-based unsupervised learning and recognition with PCM synapse circuits has been presented so far.

Here we present a novel 1T1R synapse based on PCM capable of STDP. Potentiation of the synapse is achieved via partial crystallization enabling a gradual increase of synapse conductance, while synapse depression occurs by amorphization in the reset transition. STDP characteristics are demonstrated by experiments as a function of the initial resistance state and of the number of potentiating pulses. We demonstrate the ability to learn and recognize patterns in a fully-connected neuromorphic network and we propose for the first time the input noise as a means to depress background synapses, thus enabling on-line pattern learning, forgetting and updating. Training of the PCM synapse network with alternating and multiple visual patterns according to the MNIST data base is shown. Pattern recognition with multiple layers is finally addressed for improved learning efficiency.

MATERIALS AND METHODS

PCM Characteristics

Figure 1 shows the PCM device used in this work (a) and its characteristics. The PCM was fabricated with 45 nm technology

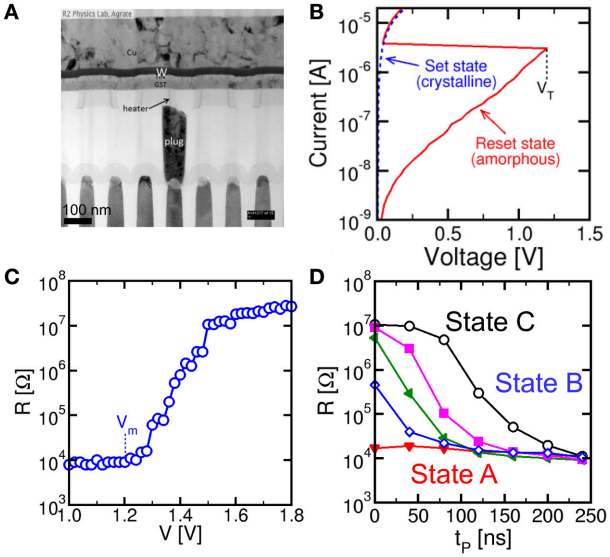


FIGURE 1 | Cross sectional view of a PCM obtained by transmission electron microscopy (TEM) (A), measured quasi-stationary I-V curves for the PCM device in the crystalline and amorphous phase (B), reset characteristic of R as a function of the write voltage for pulse-width 40 ns (C) and set characteristics of R as a function of the set pulse-width t_P and voltage V_{set} = 1.05 V for variable initial PCM state (D). The PCM device shows fast switching at low voltage, thus supporting PCM technology for low-voltage, low-power synapses in neuromorphic systems.

and consists of an active $Ge_2Sb_2Te_5$ (GST) layer between a confined bottom electrode (or heater) and a top electrode (Servalli, 2009). The PCM top electrode was made of a Cu/W/TiN multilayer connecting all cells along a row in the array, while the bottom electrode consisted of a tungsten plug and a sub-lithographic TiN heater connected to the GST layer. The active material GST is a well-known phase change material, which remains stable in 2 phases, namely the crystalline phase and the amorphous phase (Wong et al., 2010). The 2 phases differ by their respective resistance, as displayed by the I-V characteristics in **Figure 1B**: while the crystalline (set) state shows a relatively low resistance, the amorphous (reset) state shows high resistance and a typical threshold switching behavior at a characteristic threshold voltage V_T (Ielmini and Zhang, 2007). To change the PCM state, positive voltage pulses are applied between the top electrode and the heater. **Figure 1C** shows the resistance R measured after the application of a rectangular write pulse as a function of the pulse amplitude V. The PCM device was initially prepared in the set state with $R = 10\ k\Omega$ by application of a pulse with amplitude 1.2 V for 250 ns, before any applied pulse. Data show that R remains constant, until the applied voltage exceeds the voltage V_m for GST melting, causing amorphization, around 1.2 V, which corresponds to the melting voltage of the device. Above V_m, the applied pulse is able to induce melting, which leaves the GST volume in an amorphous phase as the voltage pulse is completed. The amorphous volume increases with V, thus leading to the increase of R with V in the characteristic of **Figure 1C**. To recover the initial crystalline phase, a rectangular

pulse with voltage below V_m is applied. A voltage $V_{reset} = 1.75\,V$ is sufficient to induce a resistance change to about 20 MΩ, corresponding to a full reset state. **Figure 1D** shows the resistance R measured after a set pulse with voltage $V_{set} = 1.05\,V$ as a function of the pulse-width t_P and for increasing initial R from 15 kΩ to 10 MΩ of the PCM (different colors in **Figure 1D**). In general, R decreases with increase in t_P as a result of the increased crystalline fraction (Cassinerio et al., 2013). A pulse width of about 250 ns is generally sufficient to complete crystallization within the GST layer irrespective of the initial value of R, thus supporting the good quality of PCM in terms of fast memory, low write voltage and low power consumption.

1T1R Architecture

Figure 2 schematically shows a neuron/synapse/neuron block of the neuromorphic network. Here, the synapse consists in a 1T1R structure where the PCM cell is connected in series with a MOS transistor. The transistor width and length must be suitable to drive a current around 300 μA, which is needed for set and reset transition in the PCM with 45 nm technology (Servalli, 2009). As a reference, an embedded PCM device with 1T1R structure has an area (almost equal to the transistor area) of $36F^2$, where F is the minimum feature size of the technology, for $F = 90\,nm$ and a write current of 400 μA (Annunziata et al., 2009). The 1T1R synapse has 3 terminals, namely the gate electrode of the transistor, the top electrode (TE) of the PCM and the bottom electrode consisting of the transistor channel contact not connected to the PCM. The synapse gate voltage V_G is driven by the pre-synaptic neuron (PRE), which applies a sequence of rectangular spikes. The positive gate voltage activates a current spike in the synapse which is fed into the post-synaptic neuron (POST). Each neuron in the neuromorphic network consists of a leaky integrate and fire (LIF) circuit, where the input current spike is integrated by the first stage, thus raising the internal (or membrane) potential V_{int}. The TE voltage V_{TE} is controlled by the POST, and is normally equal to a negative constant value, e.g., −30 mV. Thanks to the negative V_{TE}, a negative current spike is generated in the 1T1R in correspondence of the PRE spike, hence causing a positive increase of V_{int} in the inverting integrator of **Figure 2**. The relatively low V_{TE} ensures that the resistance state of the PCM is not changed, thus avoiding unwanted synaptic plasticity during the communication mode. The POST also controls the gate voltage of the synapse in the connection to the neuron in the next layer (not shown in **Figure 2**). Therefore, the scheme in **Figure 2** represents the building block to be replicated to achieve a generic multilayer neuromorphic array. Note finally that the 1T1R synapse in **Figure 2** can be considered a simplified version of the 2-transistor/1-resistor (2T1R) synapse presented by Wang et al. where communication and plasticity were achieved by 2 separate transistors (Wang et al., 2015), instead of only one transistor in the present solution.

As V_{int} exceeds a given threshold V_{th} of a comparator, the fire stage delivers a pulse back to the TE to update the weight of the synapse. The TE spike contains 2 rectangular pulses, the second pulse having a higher amplitude than the first one. The specific shape of the V_{TE} spike results in a change in the PCM resistance depending on the relative time delay between the PRE and POST spikes, in agreement with the STDP protocol. STDP in the PCM synapse is illustrated in **Figure 3**, showing the applied pulses from the PRE and the POST. The PRE spike is rectangular, with a 10 ms pulse-width and amplitude $V_G = 0.87\,V$, followed by a 10 ms after-pulse at zero voltage. The POST spike lasts 20 ms overall, and includes two pulses of width t_P at the beginning of the first and the second halves of the total pulse. The amplitudes of the first and second pulses are $V_{set} = 1.05\,V$ and $V_{reset} = 1.75\,V$, respectively, intercalated by wait times at zero voltage. Amplitudes V_{set} and V_{reset} are tuned to induce set transition (crystallization) and reset transition (amorphization), respectively, according to the PCM characteristics in **Figure 1**. These values should be suitably adjusted according to the specific memory technology integrated in the synapse.

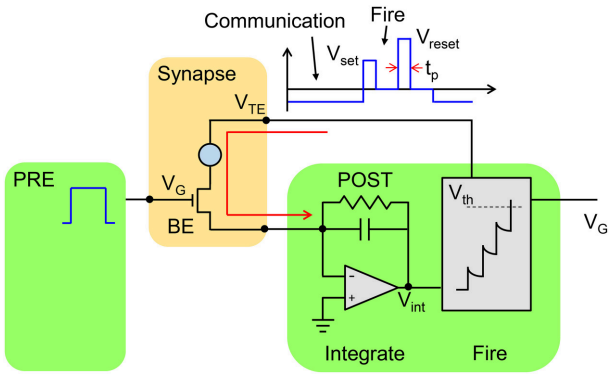


FIGURE 2 | Schematic illustration of the neuromorphic network with a 1T1R synapse. The PRE drives the MOS transistor gate voltage V_G, thus activating a current spike due to the low negative TE voltage ($V_{TE} = -30\,mV$) set by the POST. The current spikes are fed into the POST, which eventually delivers a V_{TE} spike back to the synapse as the internal voltage V_{int} exceeds a threshold V_{th}. The V_{TE} spike includes a set and reset pulse to induce potentiation/depression according to the STDP protocol.

We define the relative time delay Δt given by:

$$\Delta t = t_{post} - t_{pre},$$

where t_{post} is the initial time of the POST spike and t_{pre} is the initial time of the PRE spike, as shown in **Figure 3**. If the PRE spike appears before the POST spike (a), the relative delay Δt is positive and the PRE spike overlaps with the POST spike during the set pulse of voltage V_{set}, thus inducing set transition in the PCM with a consequent decrease of resistance. This corresponds to the so-called long-term potentiation (LTP) in the STDP protocol. If the PRE spike appears after the POST spike (b), the relative delay Δt is negative and the PRE spike overlaps with the POST spike during the reset pulse of voltage V_{reset}, thus inducing reset transition in the PCM with a consequent increase of resistance. This corresponds

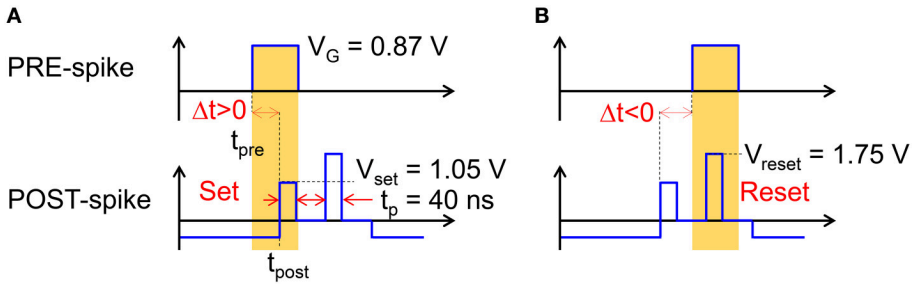


FIGURE 3 | Scheme of the applied pulses from the PRE and POST neurons to the 1T1R synapse. In the case of small positive delay Δt **(A)**, when the PRE spike is applied just before the POST spike, the PCM receives a potentiating pulse with voltage V_{set} inducing set transition. On the other hand, for small negative delay Δt **(B)**, when the PRE spike is applied just after the POST spike, the PCM receives a depressing pulse with voltage V_{reset} inducing reset transition. For positive/negative delays larger than 10 ms, there is no overlap between PRE and POST spikes, thus no potentiation/depression can take place.

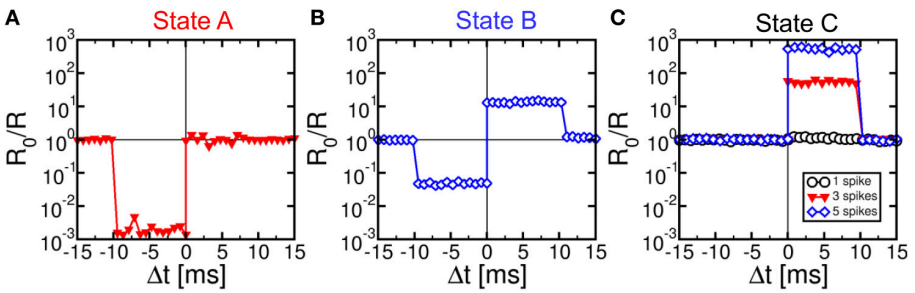


FIGURE 4 | STDP characteristics, namely measured change of conductance R_0/R as a function of delay Δt, for various PCM states, namely state A ($R_0 = 15$ kΩ), state B ($R_0 = 500$ kΩ), and state C ($R_0 = 10$ MΩ), also reported in Figure 1D. Depression and/or potentiation are shown depending on delay and initial state, providing a confirmation of the STDP capability in our 1T1R synapse.

to the so-called long-term depression (LTD) in the STDP protocol.

RESULTS

STDP Characteristics

We characterized STDP characteristics in a 1T1R synapse, obtained by wire-bonding a MOS transistor and a PCM device on 2 separate chips. The transistor size was $L = 1\,\mu$m and $W = 10\,\mu$m and the device was able to deliver sufficient current to switch the PCM device during set and reset. To demonstrate STDP operation, voltage pulses as in **Figure 3** were applied to the transistor gate and to the TE terminal with variable delay Δt and variable initial resistance R_0 of the PCM device. We used a pulse-width $t_P = 40$ ns of set/reset pulses in the POST spike, i.e., the same as in **Figures 1C,D**. **Figure 4** shows the measured change of conductance R_0/R, where R_0 and R were measured before and after the applied gate/TE pulses, for the 3 initial states of the PCM shown in **Figure 1D**, namely state A close to the full set state ($R_0 = 15$ kΩ), state B which is intermediate between set and reset states ($R_0 = 500$ kΩ), and state C close to the full reset state ($R_0 = 10$ MΩ). R was measured after one spike event in all cases except for state C, where 1, 3, and 5 spikes were used in the experiments. State A (**Figure 4A**) displays strong depression for $\Delta t < 0$, indicating a resistance increase by about 3 orders of magnitude corresponding to the full resistance window of the PCM device between set and reset states in **Figure 1C**. On the other hand, state A does not show any potentiation, since the phase is already almost completely crystallized in this state. State B (**Figure 4B**) shows both depression ($\Delta t < 0$) and potentiation ($\Delta t > 0$), since both set and reset transition are possible for this intermediate state. Finally, state C (**Figure 4C**) shows no depression, since this state is already fully amorphized. In the case of one spike, the PCM also shows no potentiation, since a 40-ns pulse is not able to induce significant crystallization in the fully-amorphized state according to the set characteristics in **Figure 1D**. Potentiation however arises after an increasing number of spikes, reaching about a factor $10^3\times$ in the case of 5 repeated spikes with the same delay. These characteristics demonstrated STDP with abrupt depression and gradual potentiation due to cumulative crystallization in the PCM device (Cassinerio et al., 2013). Note that $t_P = 40$ ns was chosen to be long enough to allow for full reset of the PCM device, while providing a partial and additive crystallization according to **Figure 1D**. A longer t_P would result in slightly different STDP characteristics, due to the larger crystallization similar to the enhanced potentiation with larger number of spikes in **Figure 4C**. On the other hand, depression would not be affected by increasing t_P, since the reset transition only depends on the quenching time.

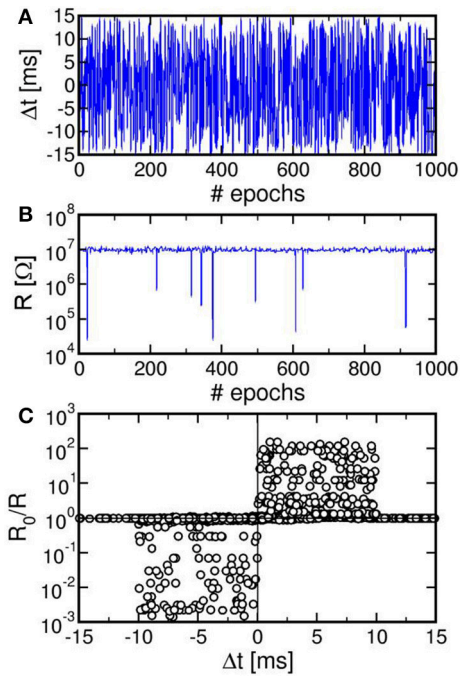


FIGURE 5 | Result of a random spiking experiment, showing the random delay Δt as a function of the epoch (A), corresponding synapse resistance as a function of the epoch (B), and correlation between Δt and R_0/R (C). The correlation between delay and conductance change is consistent with the STDP characteristics at variable resistance in **Figure 4**.

We also verified that continuous spiking with random relative delay Δt leads to random potentiation and depression of a single PCM synapse. **Figure 5** shows the results of a random Δt spiking experiment over 1000 epochs (i.e., spike events), reporting the Δt (a), the synapse resistance R as a function of the number of epochs (b), and a correlation between R_0/R and Δt (c), where R_0 and R were measured before and after each spike in the sequence. Due to the uniform distribution of Δt adopted in our experiment, R in **Figure 5B** remains close to the full reset state for most of the experiment. Only few obvious resistance drops were obtained, since at least 3 pulses with $\Delta t > 0$ are needed in **Figure 4C** to achieve potentiation from the full reset state. The correlation between Δt and R_0/R over 10^4 spikes in **Figure 5C** nicely agrees with the STDP characteristics in **Figure 4**, thus further supporting the STDP capability in our PCM–based synapse.

Note that potentiation/depression in **Figures 4**, **5** only take place during the set/reset pulses of pulse-width 40 ns, which is a negligible fraction of the spike timescale of 10 ms. This ensures that the energy consumption is negligible for synaptic plasticity as required by low power applications of the neuromorphic system.

Neuromorphic Network

Due to the simplicity of the POST spike shape including a set pulse and a reset pulse, the STDP characteristics in **Figures 4**, **5** show constant depression and potentiation for $\Delta t < 0$ and $\Delta t > 0$, respectively, in contrast to the exponential-like decay which was revealed by previous *in-vivo* experiments (Bi and Poo, 1998). In addition, STDP characteristics in **Figures 4**, **5** are affected by a large window which can reach 1000x in one single spike, as opposed to the gradual change of only few percent of biological synapses (Bi and Poo, 1998). To demonstrate that the simplified features of our STDP do not prevent a proper learning capability in our synapse, we performed simulations of pattern learning in a fully-connected perceptron with 2 neuron layers and 1T1R PCM-based synapses. **Figure 6** schematically illustrates the adopted architecture (a) and shows a practical circuit implementation with 1T1R synapses (b). The input pattern stimulates the first layer of neurons, consisting of a 28×28 retina in our simulations. Each of these 1st layer (PRE) neurons is connected to each 2nd-layer (POST) neurons via a synapse. We varied the number of POSTs in the 2nd layer and the intra-layer synaptic interaction depending on the purpose of the simulation. The 2-layer neuromorphic network can be arranged in the array-type synaptic architecture in **Figure 6B**, where a synapse in row i and column j, with $i = 1, 2, 3, \ldots, N$ and $j = 1, 2, 3, \ldots, M$, represents the connection between the i-th PRE and the j-th POST. Therefore, the generic i-th PRE drives the gate terminals of all 1T1R synapses within the corresponding row, while the generic j-th POST receives the total current generated in the j-th column of synapses and drives the TE terminals of all synapses in the j-th column, according to the scheme in **Figure 2**.

Simulation of Learning of a Single Pattern

Figure 7 shows the simulation results for the case of a 28x28 PRE retina array ($N = 784$) with a single POST ($M = 1$). Simulations were obtained with the software MATLAB and the model for PCM crystallization dynamics was obtained by interpolating data in **Figure 1D**. CMOS neuron circuitry was modeled with ideal integrators, comparators and arbitrary waveform generators, while the transistor in the 1T1R was modeled as a series resistance of 2.4 kΩ during communication and fire. The input pattern in **Figure 7A** consists of a handwritten "1" chosen within the MNIST database (LeCun et al., 1998). The pattern was randomly alternated with random noise (**Figure 7B**) for the purpose of inducing random spikes which uniformly depress all background synapses not belonging to the pattern. PRE-synaptic neurons were randomly activated during each noise event to allow for uniform depression of the background. Pattern and noise were presented with probability 50% each with clock time $t_{ck} = 10$ ms. Noise consists in the excitation of an average of 51 neurons randomly selected within the 784 PREs, corresponding to a fraction of 6.5% of neurons. During each noise epoch we extracted a different instance of white 1/0 noise. PRE spikes led to the excitation of synaptic currents that were integrated by the single POST in the 2nd layer, causing fire events every time the internal voltage exceeded V_{th}.

The evolution of the synaptic weights is shown by the color maps of conductance 1/R at $t = 0$ s (**Figure 7C**), $t = 3.5$ s (d) and $t = 7$ s (e), also corresponding to the total simulated time. We assumed that the initial distribution of weights is random between set and reset states, which can be obtained, for

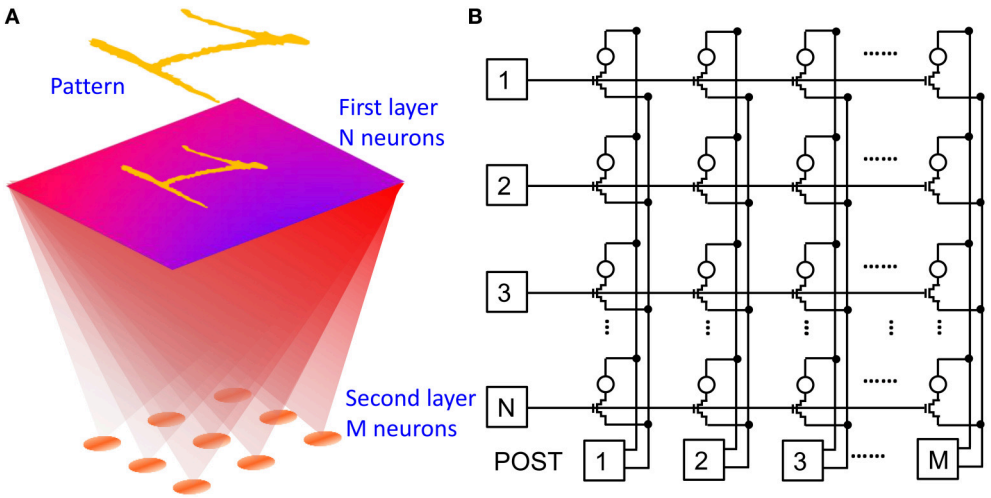


FIGURE 6 | Neuromorphic network adopted in our simulations: schematic illustration (A) and corresponding circuit (B). A first neuron layer with $N = 28 \times 28$ neurons is fully connected to a second neuron layer with M neurons through 1T1R PCM-based synapses. The first layer delivers spikes in response to presentation of one or more visual patterns. During training, STDP within the synapses leads to LTP/LTD update of the synapse weights eventually resulting in the specialization of the output neurons in recognizing the submitted patterns.

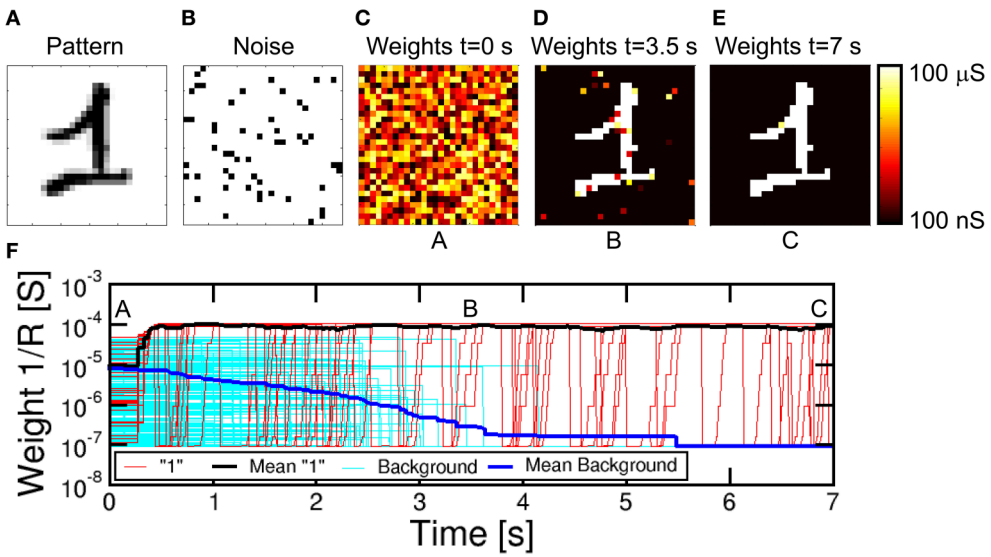


FIGURE 7 | Simulation results for pattern learning. The input pattern "1" **(A)** is presented at the input together with noise **(B)**. Synaptic weights are random at $t = 0$ s **(C)**, then they specialize at progressive times 3.5 s **(D)** and 7 s **(E)**. The corresponding complete evolution of synapse weights for increasing time is shown in **(F)**, with positions A, B, and C related to **(C–E)**. Red lines represent synapses for pattern, cyan lines are the background synapses, while the black and blue lines are the mean pattern and background synapses, showing progressive learning and specialization.

instance, by initially resetting all cells, then applying relatively short set pulse with voltage close to the PCM threshold voltage V_T. A random-set operation was shown to generate random bits in RRAM, thus enabling true random number generation (Balatti et al., 2015). **Figure 7F** shows the detailed time evolution of the synaptic weights, including 25, out of a total of 76, representative synapses within the pattern and other 236, from a total of 708, from the background, together with the corresponding average weights. Starting from the initial random distribution, the pattern weights (in red in **Figure 7F**) start to potentiate after approximately 0.3 s, reaching a value of 10^{-4} Ω^{-1} around about 0.4 s. This is the result of cumulative crystallization in the PCM as a result of multiple STDP events with $\Delta t > 0$, corresponding, e.g., to the presentation of a pattern which induces a fire in the POST. Background synapses (in cyan in **Figure 7F**) are instead depressed over a longer scale of about 3.5 s, where they reach a conductance of about 10^{-7} Ω^{-1} corresponding to the full reset state. The depression mechanism takes advantage of the

random noise appearing at the PRE neuron layer. Since noise is uncorrelated, it only causes synapse depression when the noise PRE spike comes soon after a previous fire (thus with $\Delta t < 0$) most probably induced by pattern spikes. Therefore, noise plays a key role in depression, although it should be kept to a moderate frequency and moderate density (6.5% in **Figure 7**) during training to avoid interference with stable pattern learning. Note the fast pattern learning relatively to the slow background depression, as also evidenced by the evolution of synapse weights in **Figure 7D** at 3.5 s, where depression is still not uniformly achieved in the background. The rate of background depression might be enhanced by increasing the noise density, however at the expense of a disturbed potentiation of pattern synapses. In fact, a high noise density might lead to an increased probability of noise-induced fire, which, if followed by pattern presentation, may result in the depression of pattern synapses according to STDP. Therefore, the ideal noise density should be dictated by the tradeoff between fast background depression and efficient pattern learning. The real time evolution of synapse during a representative simulation is reported in the movie M1 in the Supplementary Material. We did not implement device-to-device variability for simplicity. However, the impact should be negligible, since the network relies on the bistable device behavior rather than on the analog weight update of the synapse (Suri et al., 2013).

Energy and Power Consumption

To assess the power consumption of our synaptic network, we calculated the average dissipated energy E_{syn} and power $P_{syn} = E_{syn}/t_{ck}$ per synapse, which is shown in **Figure 8A** as a function of time during learning. The most significant contribution to energy dissipation is due to the PRE spike (communication) which induces a current spike of t_{ck} = 10 ms due to the constant $V_{TE} = -30$ mV. The dissipated energy $E_{syn,c}$ due to communication (not including fire) in a synapse is given by:

$$E_{syn,c} = t_{ck} \sum_i V_{TE}^2/(R_i + R_{MOS})/(NM),$$

where R_i is the resistance of the *i*-th synapse, R_{MOS} is the resistance of the MOS transistor in the on state, N and M are the numbers of PRE ($N = 784$ in our simulation) and POST ($M = 1$ in our simulation), respectively, and the summation is extended over all synapses that were activated by a PRE spike. In our calculations, we used a constant resistance $R_{MOS} = 2.4\,k\Omega$ for simplicity. The red filled points in **Figure 8A** show the calculated $E_{syn,c}$ due to the communication mode, reaching a peak of about 80 pJ as the pattern is presented to potentiated synapses after stable learning in the neuromorphic network. The corresponding dissipated power $P_{syn,c} = E_{syn,c}/t_{ck}$ is in the range of 8 nW. The dissipated energy is lower in the initial stages when the pattern is not yet learned, given the relatively low conductance of the pattern synapses.

Figure 8B shows the distribution of $E_{syn,c}$ due to spiking communication after consolidation of weights between $t = 4.2$ s and 7 s in **Figure 8A**. Note that there are 3 sub-distributions of $E_{syn,c}$, consisting of a high energy range (group I) due to pattern spiking and a low energy range, including a medium low sub-distribution (group II) and an extreme low sub-distribution (group III). Group II can be attributed to noise spikes exciting potentiated pattern synapses, which have large weights but only few are activated by the noise spikes. On the other hand, group III can be attributed to noise spikes exciting the background depressed synapses, thus corresponding to relatively few synapses with small weight on the average.

Figure 8A also shows the calculated $E_{syn,f}$ corresponding to the fire event, when a POST spike overlaps with the PRE spike, thus giving rise to LTP or LTD. These events generally involve a much larger V_{TE} and a larger corresponding current compared to the communication spike, since updating the PCM resistance requires set and reset transitions with significant Joule heating. On the other hand, due to the short pulse-width $t_P = 40$ ns, the

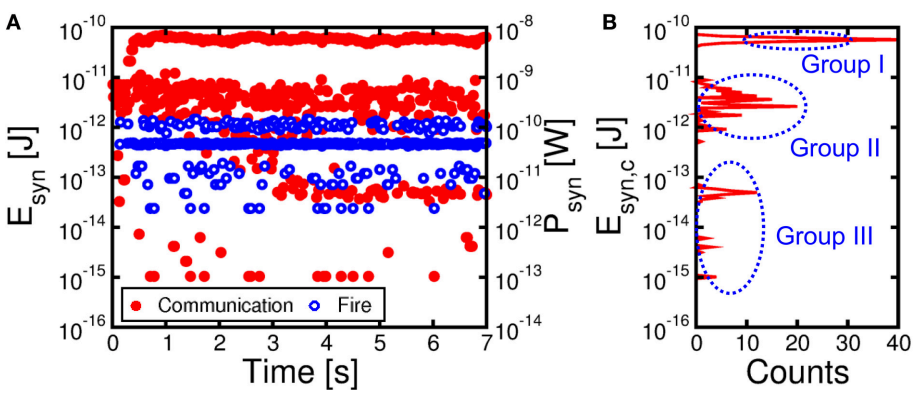


FIGURE 8 | Energy E_{syn} and mean power P_{syn} per synapse as a function of time during the learning process of Figure 7 (A) and corresponding histogram distribution of energy consumption $E_{syn,c}$ due to communication from 4.2 s to 7 s, namely after completing potentiation/depression (B). Consumption due to communication (in red) is directly induced by PRE spikes, while fire energy (in blue) corresponds to set/reset events induced by POST spikes. The energy histogram reveals 3 energy levels: Group I around 80 pJ reflects communication of pattern spikes at potentiated synapses. Group II around 5 pJ represents communication of noise spikes at potentiated pattern synapses, while group III just below 100 fJ corresponds to noise spikes at depressed background synapses.

energy dissipation is around 1 pJ, hence negligible compared to the communication energy.

Multiple Pattern Learning in Sequence or in Parallel

For on-line unsupervised pattern learning, it is important to demonstrate not only learning of a specific pattern, but also the capability to forget a previous pattern and learn a new one. The ability to reconfigure synaptic weights by learning a new pattern is in fact a key feature to rapidly interact with stimuli from a continuously-changing environment as in the real world. To verify the reconfiguration function in our neuromorphic network, we presented an input pattern to the PRE neurons for 7 s, then we presented a different pattern, where both the first and second patterns were chosen from the MNIST database. **Figure 9** shows the simulation results, including the first pattern (a), the second pattern (b), the color maps of the synaptic weights for $t = 7$ s (c), $t = 7.5$ s (d), and $t = 14$ s (e), and the synaptic conductance 1/R as a function of time (f). During the initial 7 s, pattern "1" and noise were provided with equal probabilities of 50%: the average synaptic weights show a potentiation of pattern synapse weights at 0.5 s, which is in line with **Figure 7**. At the same time, the background synapses are gradually depressed and the pattern is completely learnt after 1 s, as also shown by the weights at 7 s in **Figure 9C**. After 7 s, the input pattern is suddenly changed from "1" to "2," which causes depression of weights within pattern "1" and potentiation of weights in pattern "2." No conductance change is seen for synapses remaining in the background or pattern area. Pattern "2" is fully learned around 9 s, with depression taking slightly longer time. Sequential learning of 2 patterns is further described by movie M2 in the Supplementary Material.

We also verified the capability to learn multiple patterns in parallel, rather than in sequence as in **Figure 9**. Since a neuron can only specialize to one pattern at a time (see **Figure 9**), we extended the simulation to a network of multiple M neurons in the POST layer. **Figure 10A** shows a fully connected network including N PRE neurons and 3 POST neurons in the 2nd layer, where 3 different patterns were presented alternatively as shown in **Figure 10B**. The purpose is that each of the 3 neurons eventually specializes to a separate pattern, thus emulating the capability to recognize different patterns, such as letters, numbers, or words, by our brain. To avoid co-specialization to the same pattern, the 3 neurons were connected by inhibitory synapses, where a successful fire in any neuron leads to a partial discharge of the internal potential in all other neurons, to inhibit fire in correspondence of the same pattern and encourage specialization to other patterns. The inhibitory synapses have fixed weights, hence they can be implemented by simple resistors. The 3 input patterns in **Figure 10B** were presented with 5% probability each, with the remaining 85% consisting of noise with an average number of PRE spikes of 4 per epoch, or 0.5% of all PREs. Such low percentage of noise activity over PREs is balanced by a relatively large frequency of noise equal to 85%. After a simulated total time of 300 s, the 3 different patterns were learnt each in a different neuron, as shown by the final synaptic weights in **Figure 10C**. Decreasing the pattern presentation rate below 5% in **Figure 10** would result in a lower learning rate, while increasing the rate would cause learning instabilities. We have observed, in fact, that high pattern presentation rates cause the network to learn superposed patterns (e.g., a "1" plus a "2") or difference patterns (e.g., a "1" with the pixels of "2" excluded). This results from interaction of distinct patterns in the STDP. A low pattern rate helps reducing the probability of having interaction between different patterns.

Figure 10D shows the synaptic weights as a function of time, including the pattern weights and background weights (only synapses belonging to the background in all 3 patterns were

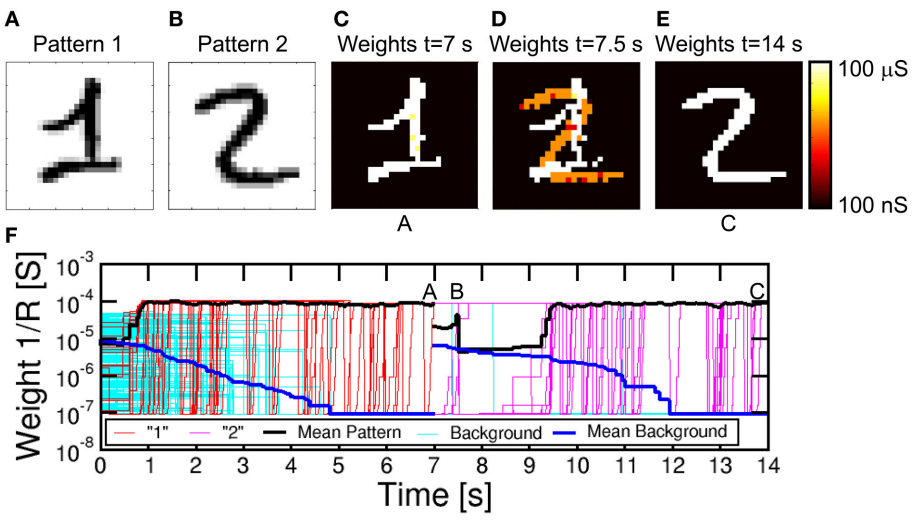


FIGURE 9 | Simulation results for pattern learning and updating. Pattern "1" and noise (A) were presented for the first 7 s, followed by pattern "2" (B) and noise for the last 7 s. After the first 7 s, in A, pattern "1" was learnt **(C)**. After starting with "2," synapses showed a mixed specialization at 7.5 s in B **(D)**, where "1" was being forgotten and "2" was being learned. Finally, at 14 s in C **(E)**, "2" was learnt. **(F)** shows the temporal evolution of synapses, with initial learning of "1," followed by updating with "2."

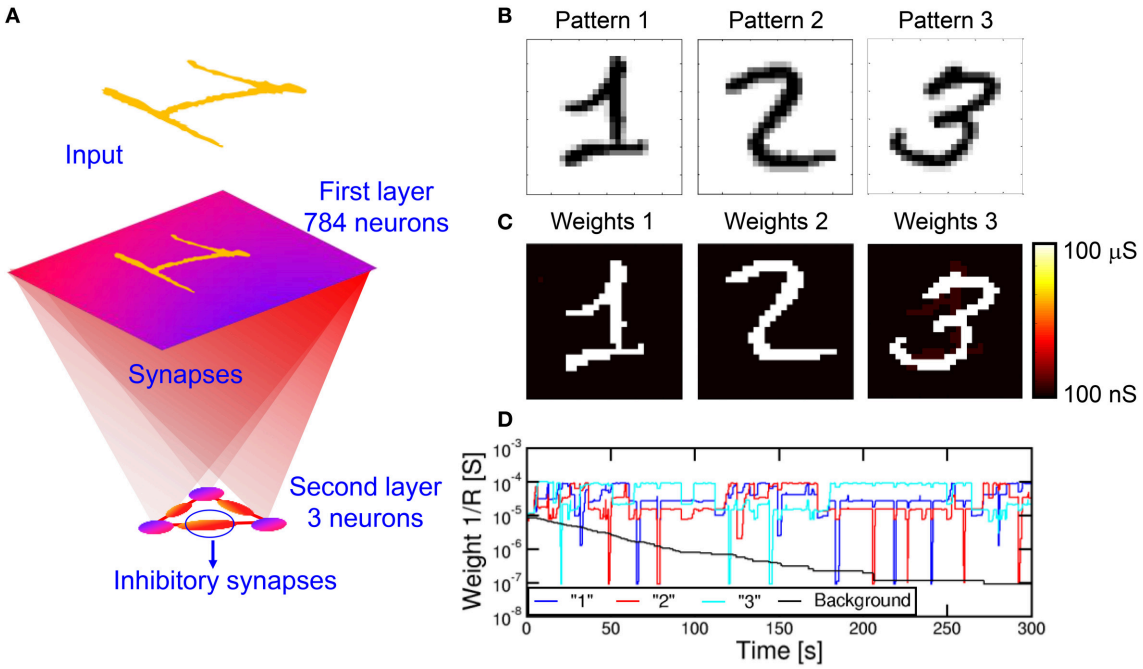


FIGURE 10 | Simulation results for multiple pattern learning. A first layer with $28 \times 28 = 784$ neurons is fully connected to three second layer neurons, each of them connected with three inhibitory synapses **(A)**. We provided three patterns "1," "2," and "3" **(B)** to the input. The three neurons specialize on different patterns **(C)**. **(D)** shows the evolution of the synapses connected to one of the post neurons, in particular the mean weight for synapses of pattern "1," "2," "3" and background. While the background gradually decreases, the learnt pattern (the highest mean conductivity) changes during time due to interference between patterns.

shown). Learning takes place in a relatively short time at the beginning of the simulation, while depression of background weights requires about 200 s due to the low activity of noise. Note also the significant oscillations of pattern weights, which are due to the instability of pattern weights due to noise. In particular, the neuron specializes on one single pattern at a time, corresponding to the highest conductance of $10^{-4}\ \Omega^{-1}$. However, the network is unable to stabilize on a single pattern due to the interference with different patterns. Nonetheless, the network is able to recognize distinct patterns in distinct POST neurons, although sometimes different POSTs learn the same input pattern. This is an unwanted effect due to the low inhibitory effect we used in the simulations, where we discharged only 20% of the capacitance of a neuron during the inhibitory action. The increase of the inhibitory factor would improve the selectivity to input patterns, although it would also cause the blockade of some POST neurons due to repeated fire in another successful POST neurons. In summary, a careful trade-off must be searched to minimize blockade events, maximize the learning efficiency and minimize the learning time. Parallel learning of 3 patterns is further described by movie M3 in the Supplementary Material.

DISCUSSION

Reducing Power Consumption via Spiking Communication

Our results support PCM devices as highly-functional synapses with learning capability and low power consumption required for the synaptic plasticity. A key limitation of the proposed scheme is however the relatively large power consumed during communication (**Figure 8**). Assuming a synapse density of 10^{11} cm^{-2} as in the human cortex, a power per synapse of 8 nW would translate in a power density of almost 1 $kWcm^{-2}$, which is comparable to a multicore CPU in conventional Von Neumann computing. The large power consumption is due to the relatively long current spike lasting 10 ms in response to the PRE spike applied to the transistor gate, where the relatively long pulse width is dictated by the STDP dynamics in the 10–100 ms time scale for real time learning and interaction (Bi and Poo, 1998). However, a spiking V_{TE} can be adopted to reduce the dissipated energy during the spike. For instance, **Figure 11** shows a spiking waveform of V_{TE}, consisting of pulses of $t_{spike} = 1\,\mu s$ width and spiking period $T_{spike} = 1$ ms, corresponding to a spiking frequency of 1 kHz and a duty cycle of 10^{-3}. The reduced duty cycle results in a reduction of power consumption by a factor 10^3, clearly bringing our neuromorphic solution in the territory of low power chips.

An additional advantage of adopting a spiking V_{TE} with low duty cycle is the ability to reduce the capacitance in the neuron integrator stage. In fact, the capacitance can be estimated by:

$$C \approx \Delta Q / V_{th},$$

where ΔQ is the integrated charge contributed by the current, equal to $\Delta Q = I\Delta t$ in the case of a constant V_{TE} as in **Figure 2**. Assuming an array of 784 PRE neurons with 10% potentiated synapses after learning, a V_{TE} of −30 mV, a resistance

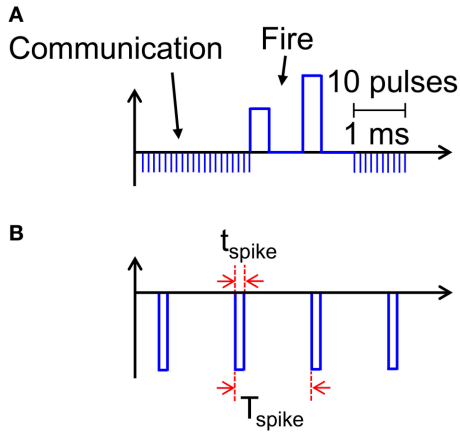


FIGURE 11 | Scheme for implementing low energy consumption communication. Instead of applying a constant $V_{TE} = -30$ mV, sequences of spikes lasting t_{spike} can allow for efficient communication **(A)**, while reducing energy and power consumption by a factor t_{spike}/T_{spike}, where T_{spike} is the time between adjacent pulses **(B)**.

of potentiated synapse of 15 kΩ, and a comparator threshold voltage $V_{th} = 0.5$ V, we obtain a capacitance of about 3 μF, which is clearly unfeasible in an integrated circuit. A duty cycle of 10^{-3} would result in a reduction of the capacitance by a factor 10^3, hence in the range of few nF. Further reduction of the power consumption and of the integrator capacitance can be obtained by reducing the duty cycle, the value of V_{TE}, and the conductivity of the PCM in the potentiated state, e.g., by adopting suitable low-conductivity phase change materials or by reducing the size of the heater controlling the cross section of the PCM device. Separation of communication and fire paths by 2T1R architecture of the synapse would allow to further reduce the current consumption and capacitor area by adopting sub-threshold bias and short pulse width of the communication gate (Kim et al., 2015; Wang et al., 2015). Finally, adopting accelerated, non-biological dynamics of tenths of ns instead of 10 ms range could allow for smaller values of integrated capacitances in the range of hundreds of fF.

Another issue consists in the wire capacitance charging energy, which is higher in the pulsing scheme. Synapses are arranged in a relatively large array, hence wires would cause a high parasitic capacitance, leading to an increase in capacitive energy dissipation in the pulsing scheme. One way to reduce the issue is to arrange synapses in a multiple smaller synapse arrays, with shorter interconnects. This approach would reduce the fan-in/fan-out of the neurons, however, with a proper design of the neuromorphic network, the issue could be acceptable, while preserving the reduction in the energy dissipation due to synapses. The capacitive energy would also be reduced by suitable voltage scaling via PCM engineering.

Multi-Layer Neuromorphic Network

To assess the learning efficiency of the neuromorphic network with PCM synapses, we performed 100 simulations of pattern learning with a total time of 2 s per each simulation. We evaluated the recognition probability P_{learn} as the number $n_{p,f}$ of fire events in the POST neuron in correspondence of the presentation of pattern "1," divided by the total number n_p of appearances of the same pattern, $P_{learn} = n_{p,f}/n_p$ (see **Figure 12A**). Similarly, we evaluated the error probability P_{err} as the number $n_{n,f}$ of POST fire events taking place in correspondence of the presentation of noise in the input (false recognitions) divided by the total number n_n of input noise appearances, $P_{err} = n_{n,f}/n_n$. Note that $n_p + n_n = n$, where n is the total number of PRE spikes within the 2 s interval of simulation. With a 2-layer network with 28 × 28 PREs and 1 POST neuron, P_{learn} was equal to 33% and P_{err} was around 6%, thus quite unsatisfactory for the purpose of on-line learning and recognition. We found that unsuccessful learning was due most of the times to depression events of pattern synapses in the case of noise causing a POST fire, followed by the presentation of the pattern in the input. In fact, PCM is particularly prone to complete depression for $\Delta t < 0$, since the reset pulse results in a large resistance increase in just one shot. After this depression event, potentiation of pattern synapses is quite difficult, since the current flowing in the depressed pattern synapse is extremely low, making a POST fire event in response to the presentation pattern quite unlikely.

To solve this issue and improve the recognition probability, we implemented a 3-layer network, as sketched in **Figure 12B**. This was done by inserting an intermediate layer with M neurons between a 28 × 28 input retina and an output layer consisting of a single neuron. All neurons between the first and the second layer were connected, and all second-layer neurons were connected to the output neuron, making the network a fully-connected architecture. The number M of neurons in the second layer was varied to study the recognition efficiency and error rates with the same pattern and noise conditions as in the calculations in **Figure 7**. **Figure 12** shows the calculated recognition probability (c) and the error probability (d) as a function of M. The recognition probability increases with M from almost 36% up to 76%, while the error rate decreases from 6 to 3%, as shown by the blue lines. The improvement is due to the compensation of synapse blockade by the additional layer, thanks to the increased number of parallel channels.

To further improve the network efficiency, we reduced the input noise from 6.5 to 5.5%. The optimized results are shown by the red curve in **Figures 12C,D**. The noise reduction leads to a slight increase in the time needed for depression of background synapses. On the other hand, the recognition efficiency increases up to 95.5% for 256 neurons in the second layer, while the error probability decreases to 0.35% in a 2 s simulation time. These results strongly support PCM-based neuromorphic chip for on-line unsupervised learning and recognition.

Impact of Noise Density on Learning Efficiency

Noise presentation alternated to the pattern allows for proper background depression and on-line unsupervised pattern updating. The randomness and non-correlation of noise allow for a general background depression and, in general, a forgetting mechanism. **Figure 13** explores more deeply the impact of

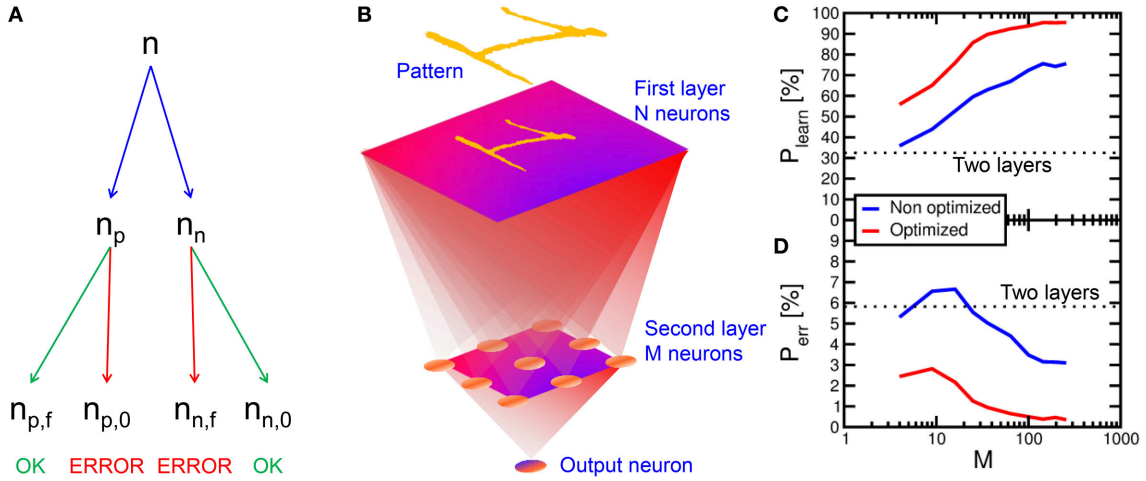


FIGURE 12 | Multi-layer simulation results. The number n of PRE spikes is composed by n_p pattern and n_n noise inputs. n_p is composed by $n_{p,f}$ (pattern leading to output spike) and $n_{p,0}$ (missing recognition). n_n is composed by $n_{n,f}$ (false recognition) and $n_{n,0}$ (absence of spike for input noise) **(A)**. After an input layer with 28×28 neurons, a second layer with variable M neurons and a third layer with one output neuron are implemented **(B)**. The recognition rate $P_{learn} = n_{p,f}/n_p$ increases with respect to the two layers network and it increases for increasing number M of second layer neurons **(C)**, while the error rate $P_{err} = n_{n,f}/n_n$ decreases **(D)**. P_{learn} further increases for optimized conditions (lower noise), reaching a 95.5% recognition, while P_{err} drops to 0.35%.

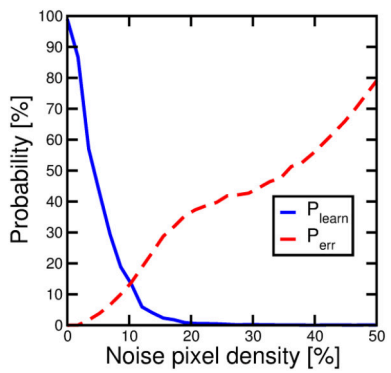


FIGURE 13 | Probability of recognizing an input pattern P_{learn}, solid line, and probability of spurious fires P_{err}, dashed line, as a function of input noise.

noise on learning efficiency. We performed pattern learning simulations as in **Figure 7**, varying the input noise density, namely the average percentage of PRE delivering a noise spike. P_{learn} shows a decrease for increasing noise density which is explained by the competition between pattern learning caused by pattern input appearance and increasing pattern forgetting induced by noise. At the same time, for increasing noise, P_{err} increases due to the increasing noise current contribution. However, note that zero noise, which seems to be the best situation, is not applicable, since background depression and pattern updating as in **Figure 9** would not be possible. Therefore, a careful trade-off between noise density and learning performance must be considered.

In conclusion, our work demonstrates PCM-based electronic synapses based on 1T1R architecture. The synapses are capable of STDP thanks to the time-dependent overlap among PRE and POST spikes in the 1T1R circuit. On-line pattern learning, recognition, forgetting and updating is demonstrated by simulations assuming the alternation of pattern and noise spikes from the PRE layer. Reduction of energy consumption and improvement of recognition efficiency are discussed with the help of simulation results. These results support PCM as promising element for electronic synapses in future neuromorphic hardware.

AUTHOR CONTRIBUTIONS

SA provided simulations of neuromorphic circuits for learning and recognition, while NC and ML contributed experimental data. All authors discussed the results and contributed to manuscript preapration. DI supervised the research.

ACKNOWLEDGMENTS

The authors are grateful to S. Balatti and Z.-Q. Wang for several discussions. This work was supported in part by the ERC Consolidator Grant No. 648635 "Resistive-switch computing Beyond CMOS."

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnins.2016.00056

REFERENCES

Ambrogio, S., Balatti, S., Nardi, F., Facchinetti, S., and Ielmini, D. (2013). Spike-timing dependent plasticity in a transistor-selected resistive switching memory. *Nanotechnology* 24:384012. doi: 10.1088/0957-4484/24/38/384012

Annunziata, R., Zuliani, P., Borghi, M., De Sandre, G., Scotti, L., Prelini, C., et al. (2009). Phase change memory technology for embedded non volatile memory applications for 90nm and beyond. *IEDM Tech. Dig.* 97–100. doi: 10.1109/iedm.2009.5424413

Balatti, S., Ambrogio, S., Wang, Z. Q., and Ielmini, D. (2015). True Random Number Generation by variability of resistive switching in oxide-based devices. *IEEE J. Emerg. Select. Topics Circ. Sys.* 5, 214–221. doi: 10.1109/JETCAS.2015.2426492

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

Bichler, O., Suri, M., Querlioz, D., Vuillaume, D., DeSalvo, B., and Gamrat, C. (2012). Visual pattern extraction using energy-efficient 2-PCM synapse neuromorphic architecture. *IEEE Trans. Electr. Dev.* 59, 2206–2214. doi: 10.1109/TED.2012.2197951

Burr, G. W., Shelby, R. M., di Nolfo, C., Jang, J. W., Shenoy, R. S., Narayanan, P., et al. (2014). "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element," in *Electron Devices Meeting (IEDM), 2014 IEEE International* (San Francisco, CA: IEEE), 29.5.1–29.5.4 . doi: 10.1109/iedm.2014.7047135

Cassinerio, M., Ciocchini, N., and Ielmini, D. (2013). Logic computation in phase change materials by threshold and memory switching. *Adv. Mat.* 25, 5975–5980. doi: 10.1002/adma.201301940

Eryilmaz, S. B.,Kuzum, D., Jeyasingh, R., Kim, S., BrightSky, M., Lam, C., et al. (2014). Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array. *Front. Neurosci.* 8:205. doi: 10.3389/fnins.2014.00205

Garbin, D., Vianello, E., Bichler, O., Rafhay, Q., Gamrat, C., Ghibaudo, G., et al. (2015). HfO_2-Based OxRAM devices as synapses for convolutional neural networks. *IEEE Trans. Electr. Dev.* 62, 2494–2501. doi: 10.1109/TED.2015.2440102

Hosseini, P., Sebastian, A., Papandreou, N., Wright, C. D., and Bhaskaran, H. (2015). Accumulation-based computing using phase-change memories with FET access devices. *IEEE Electr. Dev. Lett.* 36, 975–977. doi: 10.1109/LED.2015.2457243

Ielmini, D., and Zhang, Y. (2007). Analytical model for subthreshold conduction and threshold switching in chalcogenide-based memory devices. *J. Appl. Phys.* 102, 054517. doi: 10.1063/1.2773688

Indiveri, G., Linares-Barranco, B., Legenstein, R., Deligeorgis, G., and Prodromakis, T. (2013). Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* 24:384010. doi: 10.1088/0957-4484/24/38/384010

Indiveri, G., and Liu, S.-C. (2015). Memory and information processing in neuromorphic systems. *Proc. IEEE* 103, 1379–1397. doi: 10.1109/JPROC.2015.2444094

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297. doi: 10.1021/nl904092h

Kau, D. C., Tang, S., Karpov, I. V., Dodge, R., Klehn, B., Kalb, J. A., et al. (2009). "A stackable cross point Phase Change Memory," in *Electron Devices Meeting (IEDM), 2009 IEEE International* (Baltimore, MD: IEEE), 617–620. doi: 10.1109/IEDM.2009.5424263

Kim, S., Ishii, M., Lewis, S., Perri, T., BrightSky, M., Kim, W., et al. (2015). NVM neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with On-chip neuron circuits for continuous *in-situ* learning. *IEDM Tech. Dig.* 443.

Kuzum, D., Jeyasingh, R. G. D., Lee, B., and Wong, H.-S. P. (2012). Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Lett.* 12, 2179. doi: 10.1021/nl201040y

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Locatelli, N., Cros, V., and Grollier, J. (2014). Spin-torque building blocks. *Nat. Mater.* 13, 11–20. doi: 10.1038/nmat3823

Ohno, T., Hasegawa, T., Tsuruoka, T., Terabe, K., Gimzewski, J. K., and Aono, M. (2011). Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. *Nat. Mater.* 10, 591–595. doi: 10.1038/nmat3054

Prezioso, M., Merrikh-Bayat, F., Hoskins, B. D., Adam, G. C., Likharev, K. K., and Strukov, D. B. (2015). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521, 61–64. doi: 10.1038/nature14441

Querlioz, D., Bichler, O., Vincent, A. F., and Gamrat, C. (2015). Bioinspired programming of memory devices for implementing an inference engine. *Proc. IEEE* 103, 1398–1416. doi: 10.1109/JPROC.2015.2437616

Servalli, G. (2009). A 45nm generation Phase Change Memory technology. *IEDM Tech. Dig.* 113, 113–116. doi: 10.1109/iedm.2009.5424409

Suri, M., Bichler, O., Querlioz, D., Cueto, O., Perniola, L., Sousa, V., et al. (2011). Phase change memory as synapse for ultra-dense neuromorphic systems: application to complex visual pattern extraction. *IEDM Tech. Dig.* 79–82. doi: 10.1109/iedm.2011.6131488

Suri, M., Bichler, O., Querlioz, D., Palma, G., Vianello, E., Vuillaume, D., et al. (2012). CBRAM devices as binary synapses for low-power stochastic neuromorphic systems: auditory (Cochlea) and visual (Retina) cognitive processing applications. *IEDM Tech. Dig.* 235–238. doi: 10.1109/IEDM.2012.6479017

Suri, M., Querlioz, D., Bichler, O., Palma, G., Vianello, E., Vuillaume, D., et al. (2013). Bio-inspired stochastic computing using binary CBRAM synapses. *IEEE Trans. Electron Devices* 60, 2402. doi: 10.1109/TED.2013.2263000

Thomas, A., Niehöerster, S., Fabretti, S., Shepheard, N., Kushel, O., Kuepper, K., et al. (2015). Tunnel junction based memristors as artificial synapses. *Front. Neurosci.* 9:241. doi: 10.3389/fnins.2015.00241

Vincent, A. F., Larroque, J., Locatelli, N., Ben Romdhane, N., Bichler, O., Gamrat, C., et al. (2015). Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems. *IEEE Trans. Biomed. Circ. Syst.* 9, 166–174. doi: 10.1109/TBCAS.2015.2414423

Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., and Wong, H.-S. P. (2013). A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation. *Adv. Mater.* 25, 1774. doi: 10.1002/adma.201203680

Yu, S., Wu, Y., Jeyasingh, R., Kuzum, D., and Wong, H.-S. P. (2011). An Electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. *IEEE Trans. Electr. Dev.* 58, 2729. doi: 10.1109/TED.2011.2147791

Wang, Z. Q., Ambrogio, S., Balatti, S., and Ielmini, D. (2015). A 2-transistor/1-resistor artificial synapse capable of communication and stochastic learning in neuromorphic systems. *Front. Neurosci.* 8:438. doi: 10.3389/fnins.2014.00438

Waser, R., and Aono, M. (2007). Nanoionics-based resistive switching memories. *Nat. Mater.* 6, 833–840. doi: 10.1038/nmat2023

Wong, H.-S. P., Raoux, S., Kim, S. B., Liang, J., Reifenberg, J. P., Rajendran, B., et al. (2010). Phase change memory. *Proc. IEEE* 98, 2201–2227. doi: 10.1109/JPROC.2010.2070050

Wright, C. D., Liu, Y., Kohary, K. I., Aziz, M. M., and Hicken, R. J. (2011). Arithmetic and biologically-inspired computing using phase-change materials. *Adv. Mater.* 23, 3408. doi: 10.1002/adma.201101060

Zuliani, P., Varesi, E., Palumbo, E., Borghi, M., Tortorelli, I., Erbetta, D., et al. (2013). Overcoming temperature limitations in phase change memories with optimized $Ge_xSb_yTe_z$. *IEEE Trans. Electr. Dev.* 60, 4020–4026. doi: 10.1109/TED.2013.2285403

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 23 February 2016
doi: 10.3389/fnins.2016.00057

Emulating the Electrical Activity of the Neuron Using a Silicon Oxide RRAM Cell

Adnan Mehonic * *and Anthony J. Kenyon*

Department of Electronic and Electrical Engineering, University College London, London, UK

In recent years, formidable effort has been devoted to exploring the potential of Resistive RAM (RRAM) devices to model key features of biological synapses. This is done to strengthen the link between neuro-computing architectures and neuroscience, bearing in mind the extremely low power consumption and immense parallelism of biological systems. Here we demonstrate the feasibility of using the RRAM cell to go further and to model aspects of the electrical activity of the neuron. We focus on the specific operational procedures required for the generation of controlled voltage transients, which resemble spike-like responses. Further, we demonstrate that RRAM devices are capable of integrating input current pulses over time to produce thresholded voltage transients. We show that the frequency of the output transients can be controlled by the input signal, and we relate recent models of the redox-based nanoionic resistive memory cell to two common neuronal models, the Hodgkin-Huxley (HH) conductance model and the leaky integrate-and-fire model. We employ a simplified circuit model to phenomenologically describe voltage transient generation.

Keywords: resistive switching, neuronal dynamics, Hodgkin-Huxley, leaky integrate-and-fire, memristor

OPEN ACCESS

Edited by:
Themis Prodromakis, University of Southampton, UK

Reviewed by:
Sylvain Saïghi, University of Bordeaux, France
Hesham Mostafa, Institute for Neuroinformatics, Switzerland

***Correspondence:**
Adnan Mehonic
a.mehonic@ee.ucl.ac.uk

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *27 October 2015*
Accepted: *08 February 2016*
Published: *23 February 2016*

Citation:
Mehonic A and Kenyon AJ (2016) Emulating the Electrical Activity of the Neuron Using a Silicon Oxide RRAM Cell. Front. Neurosci. 10:57. doi: 10.3389/fnins.2016.00057

INTRODUCTION

Software models, supported by digital architecture, are convenient means to study the quantitative behavior of biological neural networks in the field of computational neuroscience. However, they cannot simulate large-scale neural systems in real time. Existing hardware, based on conventional digital logic, cannot support software that mimics detailed brain activities at a realistic scale, even with huge power consumption. Hence, artificial hardware neural systems, designed using the principles of biological neural structures, are now being developed (Indiveri, 2000; Le Masson et al., 2002; Vogelstein et al., 2008; Mitra et al., 2009). These systems are often called "neuromorphic" (Mead, 1990; Indiveri et al., 2011).

Nanodevices in which an electrical stimulus modifies electrical resistance hold great potential for a wide range of applications, the most obvious being non-volatile memories. Of such technologies, Resistive Random Access Memories (RRAMs; Waser and Aono, 2007), often classed as examples of the two-terminal elements known as memristors (Chua, 1971), are being developed as alternatives to existing memory technologies (Torrezan et al., 2011; Chen et al., 2012; Mehonic et al., 2012a). However, these devices have potential applications beyond memory, as their resistance can in some cases be semi-continuously varied, rather then being limited to binary or discrete multi-state values. Such analog variation of resistance provides a useful model of key features of the biological synapse, and RRAMs as synapses in neuromorphic circuits promise high density and efficient processing.

There have been numerous recent reports of synaptic behavior such as spike timing dependent plasticity in RRAMs (Jo et al., 2010; Indiveri et al., 2013; Yu et al., 2013; Saïghi et al., 2015). However, when it comes to modeling neuronal behavior, a hybrid approach is employed in which a RRAM/memristor models a biological synapse while CMOS circuits model neuronal dynamics. By modeling both the synapse and the neuronal electrophysiological conductance/voltage response in one device, hardware neural networks can be much simpler than existing hybrid analog/digital CMOS silicon neurons. This is the goal of the work we report here.

Here we demonstrate the feasibility of using the RRAM cell to model aspects of the electrical activity of the neuron; more specifically, the generation of voltage transients that may begin to model an action potential—neuronal spiking. Further, we demonstrate the integration capability of the device—a crucial aspect of neuronal dynamics. We discuss the operational procedures required to generate spike-like responses; we compare these spikes with those observed in biological neurons, and we relate recent models of redox-based nanoionic resistive memory cells to the conductance-based models of the neural membrane [the leaky integrate-and-fire model and the Hodgkin-Huxley (HH) model]. Although a detailed description of the physical mechanism responsible for spiking is outside the scope of this paper, we use a simple RC circuit model, similar to the one used in the leaky integrate-and-fire model, to discuss spike generation.

MATERIALS AND METHODS

Our test devices are SiO_x MIM (metal-insulator-metal) RRAM structures consisting of 37 nm-thick SiO_x layers ($x = 1.3$) sandwiched between 100 nm-thick TiN electrodes, defined by standard photolithography. Individual device sizes range from 400×400 to 5×5 μm. More details of fabrication and characterization are given elsewhere (Mehonic et al., 2015). Electrical measurements employ a Keithley Instruments 4200-SCS semiconductor parameter analyser and a Signatone probe station with 10 μm tip diameter tungsten probes. MATLAB Simulink is used for the circuit analysis.

RESULTS

More details of the resistance switching of our devices can be found in our previous study (Mehonic et al., 2015). Suffice it to say that devices require an initial abrupt electroforming step to move them from a highly insulating pristine state to a low resistance state (LRS). Subsequent resetting steps put them into a high resistance state intermediate between the LRS and pristine states. The pristine state is never recovered. Switching occurs by the formation of conductive filaments (Buckwell et al., 2015) of oxygen vacancies bridging the oxide. Devices can be cycled any times between the high and LRSs by applying the appropriate voltage or current stimuli. Transitions between states are typically fast—nanoseconds or shorter. Under unipolar operation, in which transitions from HRS to LRS and from LRS to HRS occur for the same polarity voltage stimulus, a current compliance limit is used during the HRS to LRS transition to prevent destructive breakdown of the conductive filament due to runaway Joule heating. For the opposite transition the current compliance is removed, and thermally-assisted diffusion of oxygen resets the device to the HRS.

We define two distinct classes of resistance switching: memory switching and threshold switching. The former is characterized by its non-volatility—devices remain in a specific resistance state until a stimulus causes a transition. Depending on the past history of the device, a given read voltage can result in one of two or more different currents, with the device cycled between the different states by voltage or current pulses. This is the switching mode that enables digital or multi-level operation. Threshold switching, on the other hand, is the mode in which a device is in one resistance state for low read voltages or currents, and in a different state for higher. This is a volatile system in which the measured resistance is a function of the read voltage or current.

First we examine the metastable device states that enable a fast voltage response. We explore two ways to achieve this. The first one considers typical memory switching. The second one considers threshold switching.

Generation of Controlled Voltage Transients (Voltage Spikes) Using Memory Switching

First we examine typical unipolar memory (non-volatile) switching. We obtain this type of switching by setting a higher current compliance—typically around 3 mA for our devices. The zoomed-in current-voltage curves in **Figures 1A,B** demonstrate regions of rich electrical dynamics, which are either around the transitions between the two stable states (HRS and LRS)—or regions shortly before these thresholds. Resetting (the transition from LRS to HRS) is typically gradual (**Figure 1A**), in contrast to the abrupt electroforming and setting processes. By stopping the voltage sweep at different points along this process, multi-level switching can be obtained. The end of the reset process is typically a more abrupt transition to the HRS (Mehonic et al., 2015). In the case shown, three distinct resistance states are obtained by stopping the first sweep at 2 V and the second at 3 V. Such multi-level switching is typically used to model a biological synapse. Many current spikes typically follow the overall increase of resistance.

Setting (the HRS to LRS transition) is typically an abrupt single process, although more than one level can often be observed and multi-level switching achieved (Mehonic et al., 2012b). In many cases current spikes or instabilities are observed shortly before the threshold voltage (**Figure 1B**).

We tested the generation of voltage transients (resembling voltage spikes) by applying a constant current bias to our devices and measuring the resultant voltage response. This is similar to intracellular recording from neurons using the current clamp method, tracking the generation of the action potentials. In the following text we assume that a voltage spike is an abrupt voltage increase followed by abrupt voltage decrease. More specifically, whenever the voltage increase and subsequent decrease is greater

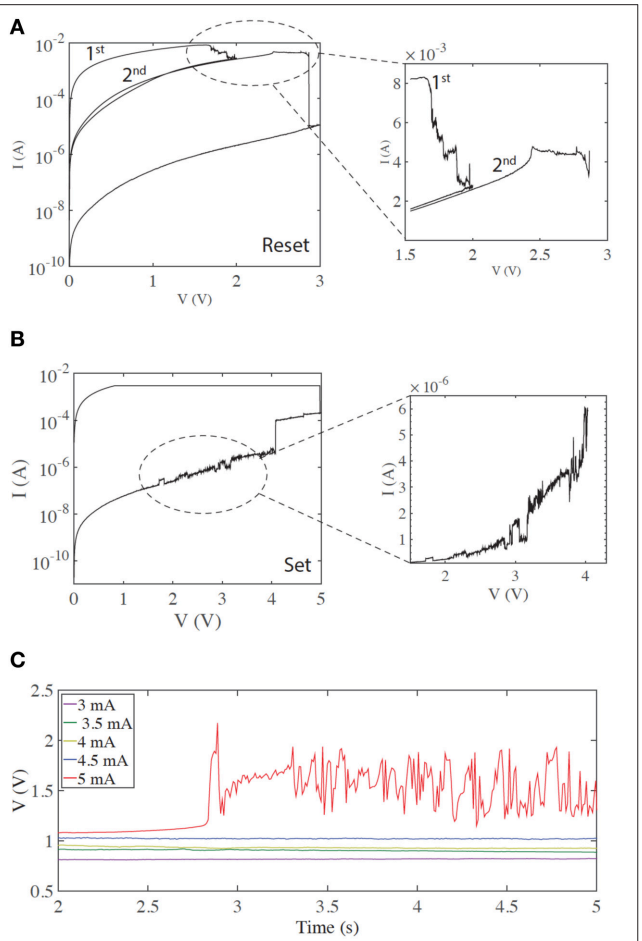


FIGURE 1 | Regions of current instabilities in I/V sweeps of SiO_x RRAM cells during non-volatile memory switching. (A) A gradual reset process, before an abrupt transition to the HRS. The zoomed region highlights the region of instability. **(B)** Current instability before an abrupt set process. **(C)** Voltage response with a constant current input, demonstrating the threshold effect of the voltage transients (spikes). In this case, no voltage transients or spikes are observed until a current of 5 mA is applied to the device.

than the standard deviation of the whole signal, and is shorter than 200 ms (typically three data points), we consider that to be a voltage spike. This is quite a relaxed definition of a voltage spike and should not be confused with the more defined stereotypical shape of the action potential generated in a biological neuron. We examined the stable, typical memory switching shown in **Figures 1A,B**, now applying a constant current and monitoring device voltage. **Figure 1C** demonstrates the resulting threshold voltage spiking/instability. Below a threshold current (here 5 mA), the voltage response is constant with no spikes. However, once the input current is above threshold significant spiking is observed. This usually occurs after some time, indicating integration of the input signal over time. Such behavior is equivalent to the neuronal generation of action potentials above a threshold input. Voltage spiking continues for a long period of time (typically >5 s) and is sometimes followed by transition to an intermediate metastable state, from which spiking resumes either spontaneously or after further increasing the input current. If current is reduced below threshold, spiking stops and a constant voltage response is recovered. A subsequent increase of input current above threshold triggers spiking again. The threshold current is usually finely defined and is approximately the same as the reset current. As the reset current is defined by current compliance during electroforming/setting (Russo et al., 2009), the threshold may be electrically tailored.

We explored the integration capability of our devices by applying a train of current pulses instead of a constant current bias. For the particular device reported here the threshold current level was around 4 mA (slightly over the 3 mA current compliance), thus we applied 4 mA excitatory current pulses (pulse width approximately 65 ms) followed by a train of 1 uA sensing pulses to track the voltage change across the device. One microampere is well below the threshold level, and hence these pulses are negligible compared to the much larger 4 mA excitatory pulses. Summing only the number of 4 mA pulses can approximate integration of the input current signal. We varied the time separation between the excitatory pulses to examine the capacity for current-time integration. Results are presented in **Figure 2**. **Figure 2A** shows the main concept of integration in the leaky integrate-and-fire model. A train of closely-spaced current pulses builds up a potential across the neural membrane until, at a specified threshold, theta, the neuron generates a voltage transient. If the separation between input current pulses is large there is a significant discharge of a membrane capacitor between the two pulses thus it takes more pulses for a voltage spike to be generated. Conversely, if pulses are more frequent the voltage spike will be generated after a fewer input pulses. We use the same analogy here, though the voltage across the device is now tracked by 1 uA sensing current pulses. **Figure 2C** shows the voltage across the device (sensed with a 1 uA current pulse) after every 4 mA excitatory pulse. The time separation between excitatory pulses is around 640 ms. A gradual build up of the voltage across the device is apparent before the voltage spike after around 35 excitatory pulses. The voltage spike is generated quicker (after fewer excitatory pulses) if the pulse separation is decreased. **Figures 2D,E** show the voltage after every excitatory pulse when the pulses are separated by 215 and 65 ms, respectively. This clearly shows the relation between the time separation between the pulses and generation of the voltage spike. This behavior is phenomenologically similar to charging and discharging of the membrane capacitor in the leaky integrate-and-fire model.

Generation of Controlled Voltage Transients (Voltage Spikes) Using Threshold Switching

In some cases devices exhibit volatile, threshold-like resistance switching, which can be initiated by using lower current compliance during the electroforming and set process. It is known that the diameter of the conductive filament produced during the electroforming step is controlled by current compliance (Ielmini, 2011; Ielmini et al., 2011). Thinner filaments, produced with lower current compliance, are less

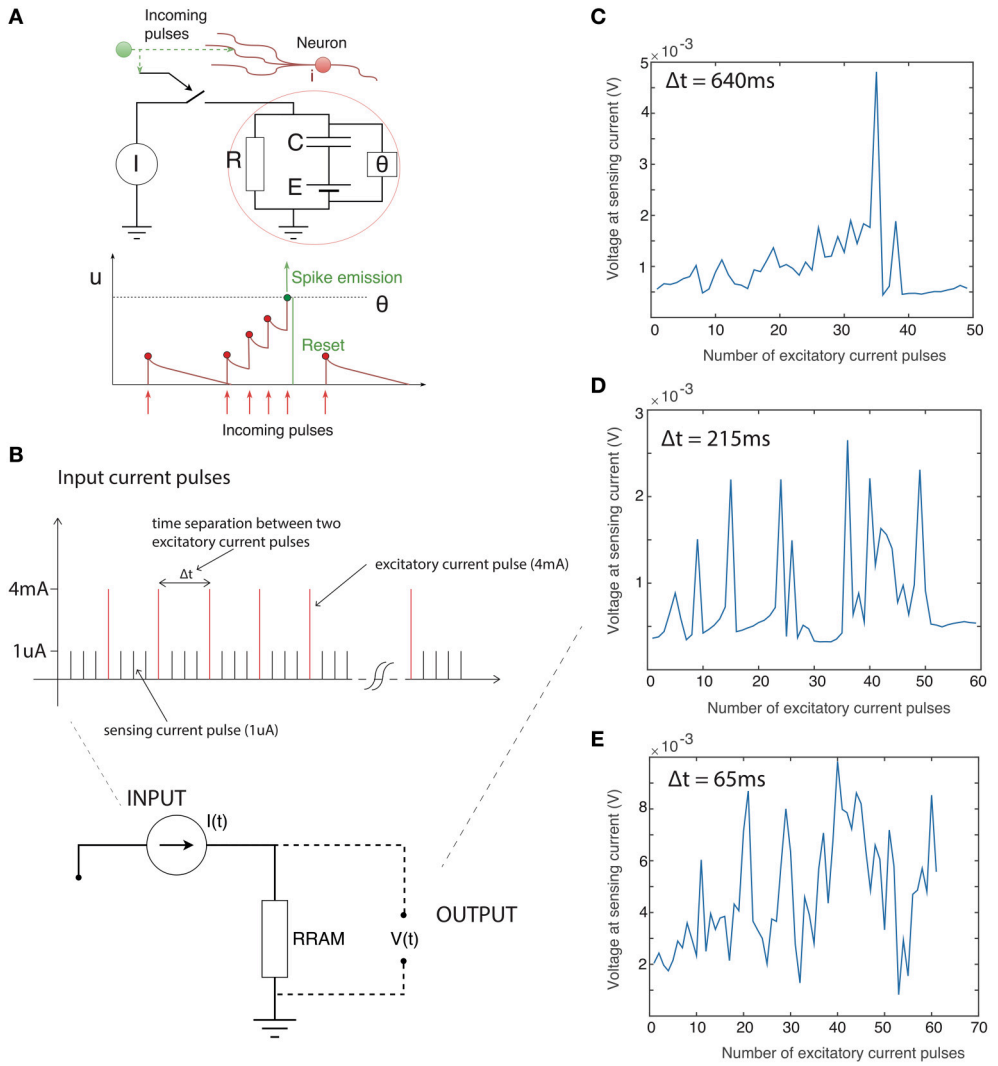


FIGURE 2 | (A) Basic representation of leaky integrate-and-fire neuronal model. Upper: schematic of model. Theta defines the voltage threshold for spiking. Lower: illustration of integration of input current pulses to generate voltage spike. X-axis is time, y axis is neuronal potential. **(B)** Time sequence of input to device: Train of excitatory current pulses (4 mA) separated by sensing current pulses (1 uA). Output of device: Voltage response measured only with the sensing 1 uA current pulses immediately after the excitatory 4 mA current pulse with the time separated of **(C)** 640 ms **(D)** 215 ms **(E)** 65 ms. The number of pulses required to be integrated decreases as the inter-pulse interval becomes shorter.

stable, and exhibit higher volatility, as seen in **Figures 3A,B**. Both states (LRS and HRS) exhibit large current instabilities.

In the case of volatile/threshold resistance switching (**Figures 3A,B**), fast spiking is observed even for lower current inputs. **Figures 3C,D** show spiking for negative currents of −1 and −2 uA, respectively. Although not fully controllable, the input current can affect the pattern of spikes. **Figure 3C** shows a chattering-like firing pattern similar to that often seen in biological neurons. **Figure 3D** shows a different firing pattern, similar to fast spiking. Although the threshold current is less finely defined than in the case of memory (non-volatile) switching, a strong correlation with the input current is evident.

Figure 4 demonstrates the effect of increasing input current from 1 to 13 uA. Less prominent firing is observed at lower currents, while the firing frequency is increased by raising the current. This is a signature of a neuronal response.

Firing events are not fully random. There is a clear pattern of a fast firing sequence followed by a refractory period of no firing. To further study this behavior we analyzed the dynamics of the firing pattern. **Figures 5A,C,E** show the firing patterns of three different input currents (1, 7, and 13 uA respectively). **Figures 5B,D,F** show the corresponding Fourier transform of the signals. It is apparent that for all three signals there are two dominant frequencies (a first peak in region 4–5 Hz and a second peak in region of 40–50 Hz). This behavior is similar for all signals shown in **Figure 4**. **Figure 5G** demonstrates an increase in the number of peaks (proportional to an average firing frequency) with increased input current.

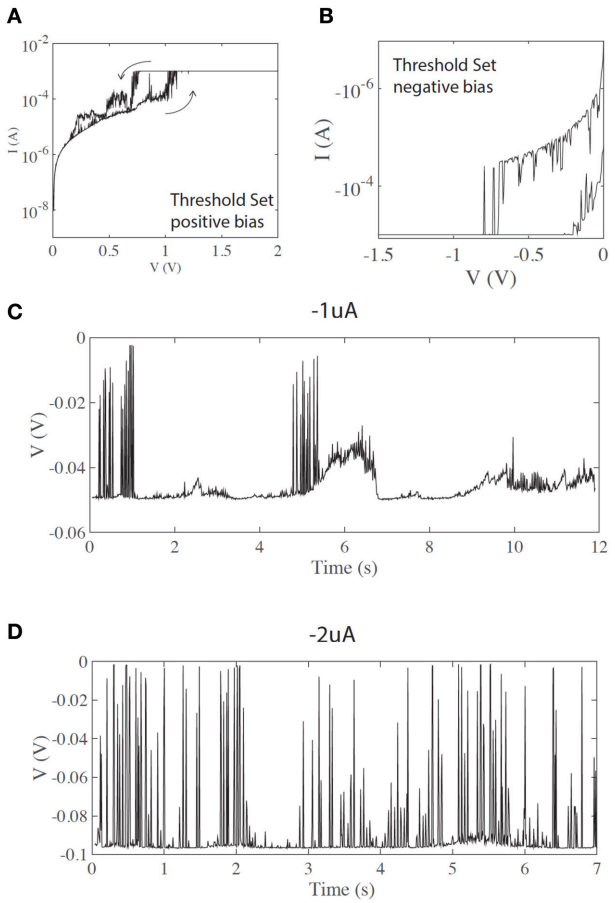


FIGURE 3 | (A,B) threshold (volatile) switching in positive and negative bias, respectively. **(C)** Short voltage spikes are observed even at lower current inputs (negative current input). **(D)** Same as **(C)** under increased current input, showing a higher count of voltage transients.

Regardless of the input current, the overall spiking pattern, resembling a chattering pattern, stays unchanged.

Generation of voltage transients (spikes) using threshold switching is less controlled than when using memory switching, although some level of control (firing frequency) is still retained. However, this approach has certain advantages. Voltage spikes are typically more pronounced and the overall operational energy is significantly lower then for the first approach using memory switching (currents of a few microAmps are sufficient to generate voltage spikes). On the other hand, on-volatile operation provides very good control of the threshold levels as well as integration of the input signal.

DISCUSSION

Comparison of the Extended Memristor Model of the ReRam System with the Hudgkin-Huxley and Leaky-and-Integrate Neuronal Models

A detailed description of the switching mechanism can be found in our previous work, though we note here that it falls within the description of redox-based nanoionic resistive memories (Waser et al., 2012; Mehonic and Kenyon, 2015). Here we will discuss the similarities and differences between the biological system described by the HH model and leaky integrate-and-fire model, the extended memristor model of ReRAM system, and our device. Schematic representations of the two systems are shown in **Figures 6A,B**. We first compare the latest redox-based nanoionic model of resistance switching (Valov et al., 2013) with the conduction-based Hodgkin-Huxley model of the neuron (Hodgkin and Huxley, 1952). The easiest way to analyse the similarities is to compare the two equivalent electric circuits. The nanoionic model takes into account the non-equilibrium states inside the memory cell and the generation of an internal electromotive force (V_{emf}) by the movement of ions during electrical biasing. This requires an expansion of memristor theory to include a nanobattery; the resultant equivalent circuit is shown in **Figure 6D**. This is the extended memristance model.

The Hodgkin-Huxley model provides an electrical description of the generation of the action potential. A set of differential equations describes the conductance of the neuron membrane, with the equivalent circuit shown in **Figure 6C**. It assumes two ionic channels (usually sodium and potassium) and one nonspecific leakage channel, as well as corresponding ion pumps. Changes in the membrane potential and in the conductivity of the ion channels generate the action potential. The model is summarized by Equation (1). The ion currents on the right-hand side are sodium, Na^+, potassium, K^+, and the leakage current. When the ion channels are fully open they have maximum conductances g_{Na}, g_K, respectively. The dynamics of the variable conductivity are defined by the gating variables n, m and h, which model ion channel opening. A generalized gating variable x is defined by a differential equation (Equation 2), with both steady state gating variable x_0 and time constant τ_x dependent on voltage u. Since there is a build up of the Nernst potential across the membrane for every ionic species, there are additional battery elements. These are modeled by E_{Na}, E_K, and E_L.

$$\sum\nolimits_k I_k = g_{Na} m^3 h\,(u - E_{Na}) + g_K n^4\,(u - E_K) + g_L\,(u - E_L) \quad (1)$$

$$\frac{dx}{dt} = -\frac{x - x_0\,(u)}{\tau_x\,(u)} \quad (2)$$

The circuit representation of the HH model is very similar to that of the Extended Memristor Model (EMM; **Figures 6C,D**). Both include a capacitance in parallel with one or more variable resistors and internal emf sources. Unsurprisingly, the EMM can be described by a similar set of equations to those of the HH model, including contributions from ionic and electrical currents and a built-in emf (Equation 3).

$$I = I_{ion}\left(V_{emf}, u\right) + I_{el}\,(x, u) = G\,(x, u) \times \left(u - t_{ion} V_{emf}\right) \quad (3)$$

With ionic current I_{ion} and electronic current I_{el}. The former is defined by the nanobattery, V_{emf}. The latter is controlled by state-dependent x. G is the conductance, u is applied voltage, and t_{ion} is the transference number (the total ionic transfer number). More details and a derivation of the model can be found in Valov et al. (2013).

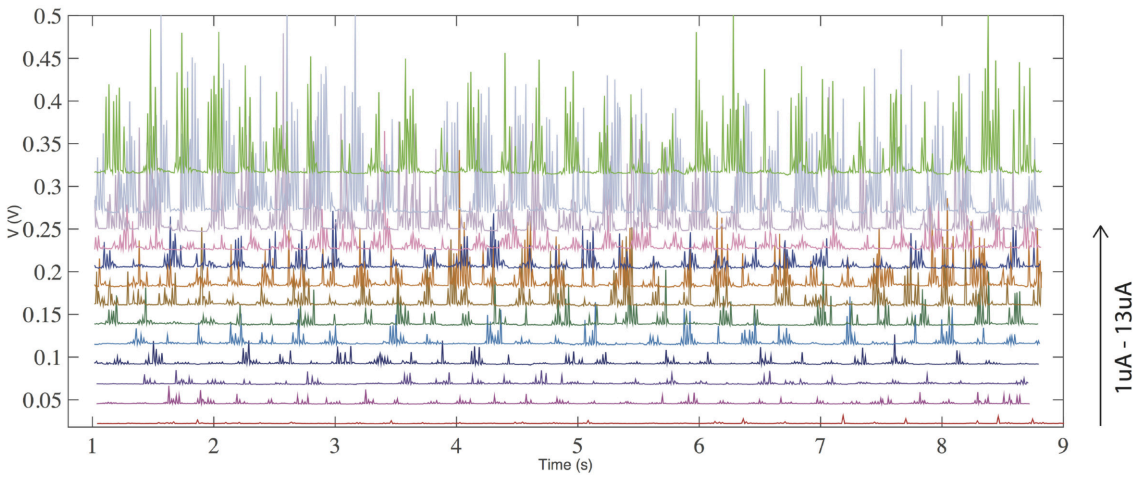


FIGURE 4 | (color online) Voltage response with a constant current input for threshold (volatile) switching. The frequency of spiking/firing is increased with an increase of the input current.

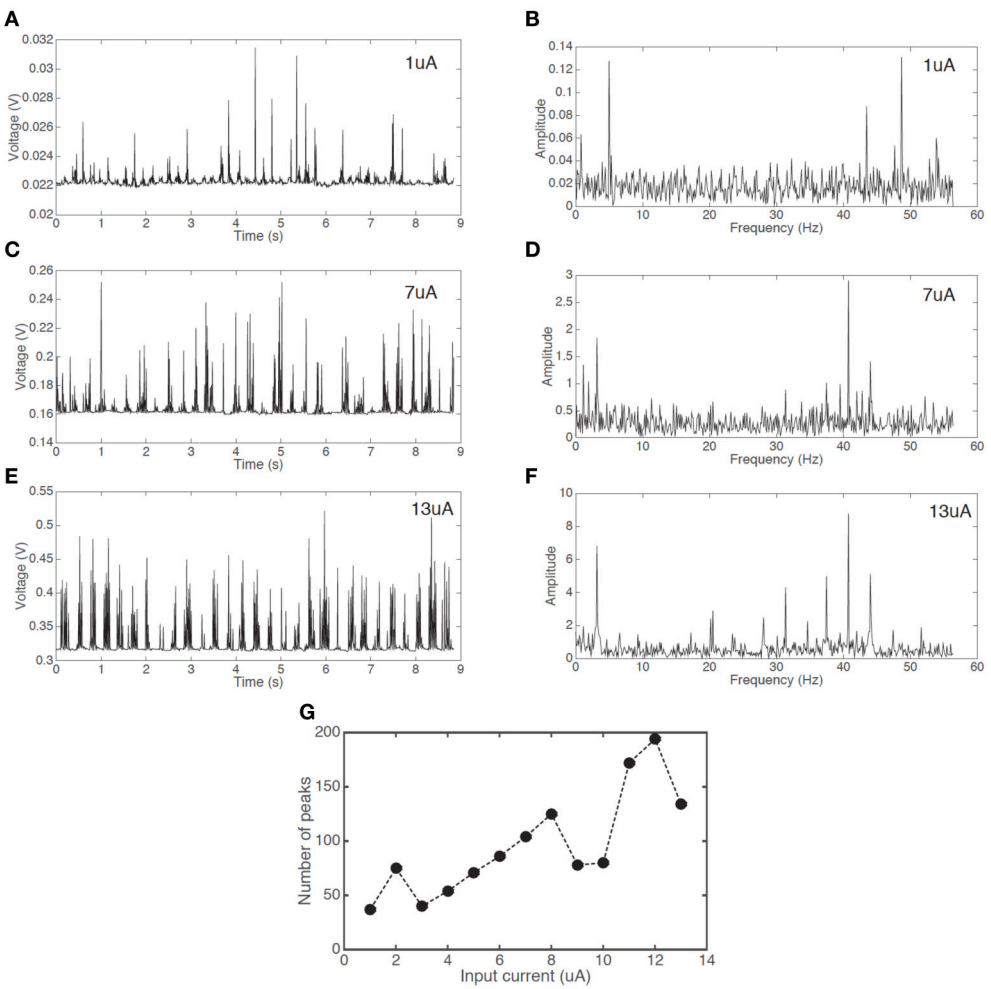


FIGURE 5 | Voltage responses with a constant current input and the corresponding Fourier transforms. Spiking signal with an input current of **(A)** 1 uA **(C)** 7 uA **(E)** 13 uA, and Fourier transform signal with the input current of **(B)** 1 uA **(D)** 7 uA **(F)** 13 uA. **(G)** The increase in the number of peaks in an interval of 8.8 s with increasing input current.

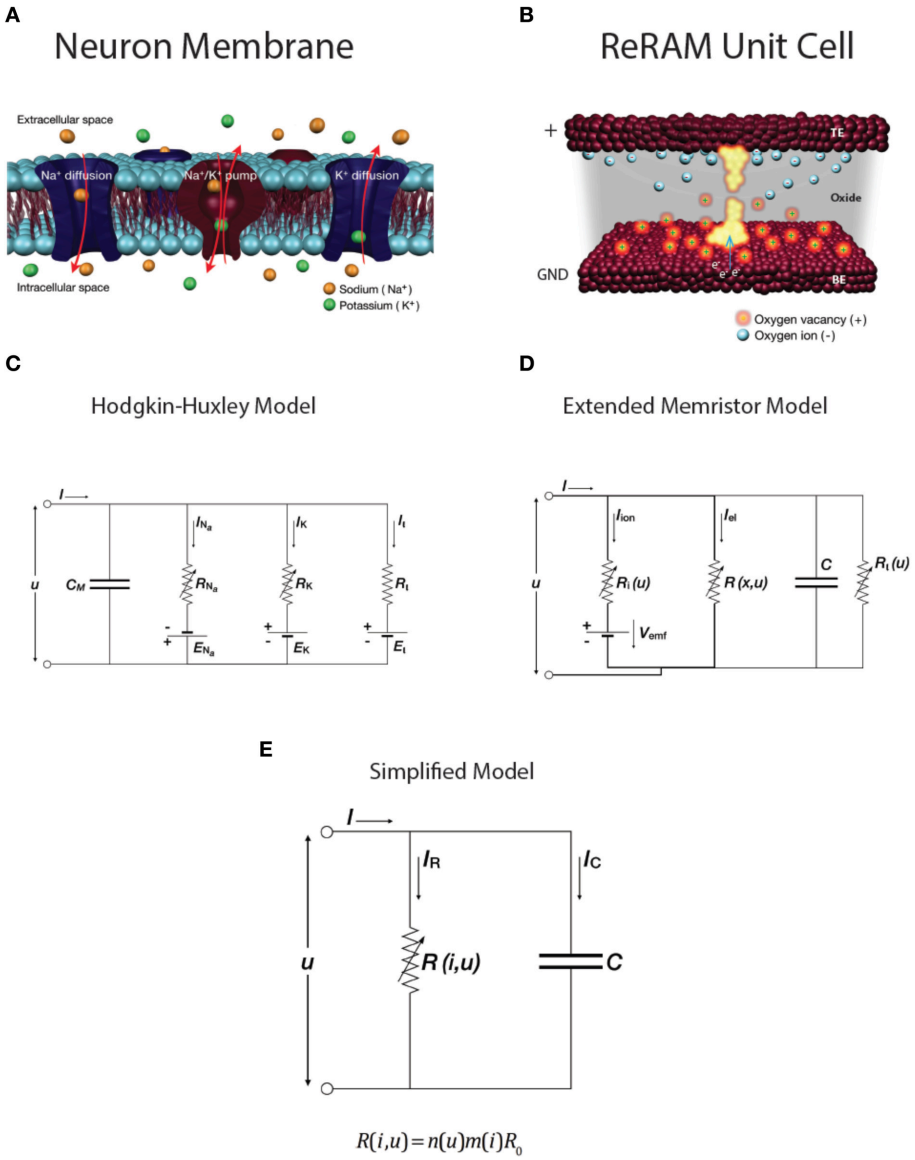


FIGURE 6 | (color online) Schematics of (A) a neuron cell membrane and (B) a ReRAM unit cell. Equivalent circuits of **(C)** Hodgkin-Huxley conductance-based model of neuron membrane and **(D)** extended memristive element. **(E)** Simplified RC model with variable resistance R.

Importantly for our discussion this V_{emf} is very small in the case of Valence Change Memory systems such as our SiO_x devices (Valov et al., 2013). This contribution is further reduced when the device is in the LRS. Similarly, the ionic resistance, R_i, is very large compared to the electronic resistance R. We may therefore make a useful simplification to the equivalent circuit model, shown in **Figure 6E**, which includes a single variable resistance.

Phenomenological Modeling of the Dynamics of a Non-Volatile SiO_x RRAM Device

To analyse the dynamics of our SiO_x RRAM system, more specifically to phenomenologically describe the generation of voltage transients, and to make comparison with neuronal dynamics, we consider the simplified model in **Figure 6E**. It is worth noting that a simple RC circuit is used in the leaky integrate-and-fire neuron models to integrate the input signal. In these models, the RC circuit does not generate any voltage spikes, but it provides a measure of voltage increase across the membrane (membrane capacitor) and when the threshold voltage is reached a separate external circuit is used to generate a voltage spike. After this voltage spike is generated the voltage across the RC circuit is reset. In contrast, in our model we do not use additional circuit elements to generate spikes; instead we examine the effect of the dynamically variable resistance *R*. Resistance is a general function of both the applied voltage and the passing current. This is similar to the HH model, in which

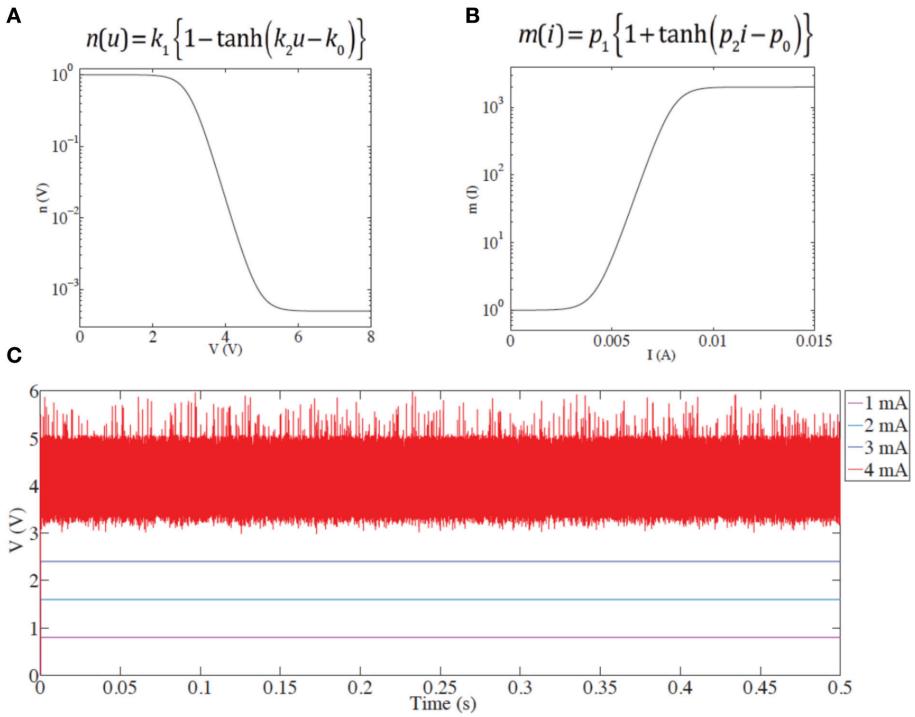


FIGURE 7 | (A) Setting coefficient dependence on applied voltage. **(B)** Resetting coefficient dependence on current. In both cases, the functional forms phenomenologically describe changes in resistance in response to applied voltages and currents. The resistance change is immediate; time evolution is not taken into account in this model. **(C)** Demonstration of the instability/spiking threshold. Below threshold (i < 4 mA), no spikes are seen. Above threshold, multiple transients result from the competition between set and reset processes, governed by *n(u)* and *m(i)* acting in opposition. The spikes are of qualitative nature and do not describe timing.

ion channel conductance is dynamically controlled by the voltage across the neural membrane. Consequently, to model voltage spike generation in our device (using non-volatile memory switching) we use some elements of both the HH model (voltage controlled resistance R) and the leaky integrate-and-fire neuronal model (RC equivalent circuit).

Although resistance transitions are controlled both by electric field and associated Joule heating, in the case of unipolar switches the set process is triggered predominantly by the electric field (voltage), while Joule heating (current) triggers the reset. To a good approximation this means that, above a certain value, current breaks the filament and increases the overall resistance, while the voltage restores the filament and reduces the resistance. This is modeled phenomenologically by two variable coefficients: the setting coefficient $n(u)$, and the resetting coefficient $m(i)$, which are phenomenologically similar to the gating coefficients in the HH model. R_0 is the previous steady state resistance. The two coefficients, $n(u)$ and $m(i)$, do not have a deeper physical meaning, but they do qualitatively describe the resistance increase with current increase and the resistance decrease with voltage increase above the threshold.

$$R(i, u) = n(u)\, m(i) R_0 \tag{4}$$

We use this circuit model to probe the origin of voltage spiking. The input current is kept constant, and the dynamics of the device voltage are observed. For the sake of simplicity and convenience we choose two continuous functions of the following form to model the coefficients $n(u)$ and $m(i)$:

$$n(u) = k_1 \left\{1 - tanh\left(k_2 u - u_0\right)\right\} \tag{5}$$

$$m(i) = p_1 \left\{1 + tanh\left(p_2 i - i_0\right)\right\} \tag{6}$$

where k_1, k_2, p_1, p_2 are unitless fitting parameters and u_0, i_0 are fitting parameters related to the thresholds of voltage and current governing setting and resetting, respectively. The functional shapes of the two coefficients are shown in **Figures 7A,B**.

Results from the above model are shown in **Figure 7C**. Voltage transients are observed only when the input current reaches a level of 4 mA. In our previous work (Mehonic et al., 2012a) we have discussed competition between the set and reset processes during constant voltage bias. Similar dynamics occur under current bias. If the initial state is the LRS and the input current is high enough to trigger a reset, this will drive the device toward the HRS. Consequently, device resistance will increase, as will the voltage drop across the device. For a constant current, the voltage will increase enough to trigger the set process, putting the device back to LRS and the whole process starts again. This competition between set and reset processes, generates voltage transients.

We note that in our model we do not assume any time dependence of the two coefficients, $n(u)$ and $m(i)$. Equations (5) and (6) do not include any time-dependent dynamics.

Furthermore, the equations are of zero-order (changes of *n(u)* and *m(i)*, and resistance are assumed to be instantaneous). This means that the model cannot provide frequency or shape analysis of the voltage responses. Instead, the aim of the model is to phenomenologically describe voltage transients and the threshold effect. In most cases the generated transients resemble a noise-like signal (a consequence of zero order dynamics) and are likely a function of simulation step size. There is therefore no correlation between the firing frequency of the experimental result in **Figure 1C** and model results in **Figure 7C**. To include shape and the frequency analysis, coefficients *n(u)* and *m(i)* should be modeled by similar differential equations to those used for the gating coefficients *x(u)* in the HH model taking into account non-zero order dynamics and the time dependency. However, the exact relation between the resistance change and applied voltage/current in RRAM systems is not yet fully established and is outside the scope of this manuscript. Nevertheless, our model clearly demonstrates the threshold effect and generation of voltage instability, without considering time evolution.

The whole discussion above considers only memory switching. Volatile/threshold, less-stable switching provides rapid resistance variations without a finely defined threshold. We suspect that rapid resistance variations are the effect of trapping/detrapping processes or random telegraph noise (RTN) affected by the redistribution of oxygen vacancies, as discussed in Balatti et al. (2014), Choi et al. (2014). The rate of movement of oxygen vacancies is increased by increasing the current input. Consequently, we observe in volatile systems that the firing frequency is also increased—a typical neuronal response.

CONCLUSION

To summarize, we have demonstrated the feasibility of using the SiO_x RRAM cell to model aspects of the voltage spiking activity of a biological neuron. This is a different approach from conventional synaptic modeling using RRAM devices. We elaborate the specific metastable device states required for the generation of voltage spiking, and demonstrate a dynamic voltage response to a constant input current and to a current pulse train. We discuss observation of threshold spiking as well as an increase of firing frequency with increased input current. We demonstrate the integration capability of our device. We compare the model of redox-based nanoionic resistive memory to the Hodgkin-Huxley neuron model and the leaky integrate-and-fire model. We use circuit simulations to further explain the voltage response. This study could provide a novel way of using RRAM devices in neuromorphic systems beyond the already-demonstrated capability to model a functional synapse.

AUTHOR CONTRIBUTIONS

AM conceived the study, performed the measurements, and wrote the initial draft of the paper. AK oversaw the project and revised the manuscript.

ACKNOWLEDGMENTS

We acknowledge financial support form the Engineering and Physical Sciences Research Council (EPSRC).

REFERENCES

Balatti, S., Ambrogio, S., Cubeta, A., Calderoni, A., Ramaswamy, N., and Ielmini, D. (2014). "Voltage-dependent random telegraph noise (RTN) in HfOx resistive RAM," in *Reliability Physics Symposium, 2014 IEEE International* (Waikoloa, HI: IEEE), MY-4. doi: 10.1109/IRPS.2014.6861159

Buckwell, M., Montesi, L., Hudziak, S., Mehonic, A., and Kenyon, A. J., (2015). Conductance tomography of conductive filaments in intrinsic silicon-rich silica RRAM. *Nanoscale* 7, 18030. doi: 10.1039/C5NR04982B

Chen, H. Y., Yu, S., Gao, B., Huang, P., Kang, J., and Wong, H. S. P. (2012). "HfOx based vertical resistive random access memory for cost-effective 3D cross-point architecture without cell selector," in *Electron Devices Meeting (IEDM), 2012 IEEE International* (San Francisco, CA), 20.7.1–20.7.4. doi: 10.1109/iedm.2012.6479083

Choi, S., Yang, Y., and Lu, W. (2014). Random telegraph noise and resistance switching analysis of oxide based resistive memory. *Nanoscale* 6, 400–404. doi: 10.1039/C3NR05016E

Chua, L. O. (1971). Memristor-the missing circuit element. *Circuit Theory IEEE Trans.* 18, 507–519. doi: 10.1109/TCT.1971.1083337

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544. doi: 10.1113/jphysiol.1952.sp004764

Ielmini, D. (2011). Modeling the universal set/reset characteristics of bipolar RRAM by field-and temperature-driven filament growth. *Electron Devices IEEE Trans.* 58, 4309–4317. doi: 10.1109/TED.2011.2167513

Ielmini, D., Nardi, F., and Cagli, C. (2011). Physical models of size-dependent nanofilament formation and rupture in NiO resistive switching memories. *Nanotechnology* 22:254022. doi: 10.1088/0957-4484/22/25/254022

Indiveri, G. (2000). Modeling selective attention using a neuromorphic analog VLSI device. *Neural Comput.* 12, 2857–2880. doi: 10.1162/089976600300014755

Indiveri, G., Linares-Barranco, B., Legenstein, R., Deligeorgis, G., and Prodromakis, T. (2013). Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* 24:384010. doi: 10.1088/0957-4484/24/38/384010

Indiveri, G., Linares-Barranco, B., Hamilton, T. J., Van Schaik, A., Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5:73. doi: 10.3389/fnins.2011.00073

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301. doi: 10.1021/nl904092h

Le Masson, G., Renaud-Le Masson, S., Debay, D., and Bal, T. (2002). Feedback inhibition controls spike transfer in hybrid thalamic circuits. *Nature* 417, 854–858. doi: 10.1038/nature00825

Mead, C. (1990). Neuromorphic electronic systems. *Proc. IEEE* 78, 1629–1636. doi: 10.1109/5.58356

Mehonic, A., Buckwell, M., Montesi, L., Garnett, L., Hudziak, S., Fearn, S., et al. (2015). Structural changes and conductance thresholds in metal-free intrinsic SiOx resistive random access memory. *J. Appl. Phys.* 117, 124505. doi: 10.1063/1.4916259

Mehonic, A., Cueff, S., Wojdak, M., Hudziak, S., Jambois, O., Labbé, C., et al. (2012a). Resistive switching in silicon suboxide films. *J. Appl. Phys.* 111, 074507. doi: 10.1063/1.3701581

Mehonic, A., Cueff, S., Wojdak, M., Hudziak, S., Jambois, O., Labbé, C., et al. (2012b). Electrically tailored resistance switching in silicon oxide. *Nanotechnology* 23:455201. doi: 10.1088/0957-4484/23/45/455201

Mehonic, A., and Kenyon, A. J. (2015). "Resistive switching in oxides," in *Defects at Oxide Surfaces*, eds J. Jupille and G. Thornton (Basel: Springer International Publishing), 401–428. doi: 10.1007/978-3-319-14367-5_13

Mitra, S., Fusi, S., and Indiveri, G. (2009). Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI. *Biomed. Circuits Syst. IEEE Trans.* 3, 32–42. doi: 10.1109/TBCAS.2008.2005781

Russo, U., Ielmini, D., Cagli, C., and Lacaita, A. L. (2009). Self-accelerated thermal dissolution model for reset programming in unipolar resistive-switching memory (RRAM) devices. *Electron Devices IEEE Trans.* 56, 193–200. doi: 10.1109/TED.2008.2010584

Saïghi, S., Mayr, C. G., Serrano-Gotarredona, T., Schmidt, H., Lecerf, G., Tomas, J., et al. (2015). Plasticity in memristive devices for spiking neural networks. *Front. Neurosci.* 9:51. doi: 10.3389/fnins.2015.00051

Torrezan, A. C., Strachan, J. P., Medeiros-Ribeiro, G., and Williams, R. S. (2011). Sub-nanosecond switching of a tantalum oxide memristor. *Nanotechnology* 22:485203. doi: 10.1088/0957-4484/22/48/485203

Valov, I., Linn, E., Tappertzhofen, S., Schmelzer, S., Van den Hurk, J., Lentz, F., et al. (2013). Nanobatteries in redox-based resistive switches require extension of memristor theory. *Nat. Commun.* 4, 1771. doi: 10.1038/ncomms2784

Vogelstein, R. J., Tenore, F., Guevremont, L., Etienne-Cummings, R., and Mushahwar, V. K. (2008). A silicon central pattern generator controls locomotion *in vivo*. *Biomed. Circuits Syst. IEEE Trans.* 2, 212–222. doi: 10.1109/TBCAS.2008.2001867

Waser, R., and Aono, M. (2007). Nanoionics-based resistive switching memories. *Nat. Mater.* 6, 833–840. doi: 10.1038/nmat2023

Waser, R., Bruchhaus, R., and Menzel, S. (2012). "Redox-based resistive switching memories," in *Nonoelectronics and Information Technology, 3rd Edn.* ed R. Waser (Weinheim: Wiley VCH), 685–710. doi: 10.1166/jnn.2012.6652

Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., and Wong, H. S. P. (2013). A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation. *Adv. Mater.* 25, 1774–1779. doi: 10.1002/adma.201203680

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 06 January 2016
doi: 10.3389/fnins.2015.00484

Energy Scaling Advantages of Resistive Memory Crossbar Based Computation and Its Application to Sparse Coding

Sapan Agarwal[1], Tu-Thach Quach[2], Ojas Parekh[3], Alexander H. Hsia[1], Erik P. DeBenedictis[3], Conrad D. James[1], Matthew J. Marinella[1] and James B. Aimone[3]**

[1] Microsystems Science and Technology, Sandia National Laboratories, Albuquerque, NM, USA, [2] Sensor Exploitation, Sandia National Laboratories, Albuquerque, NM, USA, [3] Center for Computing Research, Sandia National Laboratories, Albuquerque, NM, USA

The exponential increase in data over the last decade presents a significant challenge to analytics efforts that seek to process and interpret such data for various applications. Neural-inspired computing approaches are being developed in order to leverage the computational properties of the analog, low-power data processing observed in biological systems. Analog resistive memory crossbars can perform a parallel read or a vector-matrix multiplication as well as a parallel write or a rank-1 update with high computational efficiency. For an $N \times N$ crossbar, these two kernels can be $O(N)$ more energy efficient than a conventional digital memory-based architecture. If the read operation is noise limited, the energy to read a column can be independent of the crossbar size ($O(1)$). These two kernels form the basis of many neuromorphic algorithms such as image, text, and speech recognition. For instance, these kernels can be applied to a neural sparse coding algorithm to give an $O(N)$ reduction in energy for the entire algorithm when run with finite precision. Sparse coding is a rich problem with a host of applications including computer vision, object tracking, and more generally unsupervised learning.

Keywords: resistive memory, memristor, sparse coding, energy, neuromorphic computing

OPEN ACCESS

Edited by:
Themis Prodromakis, University of Southampton, UK

Reviewed by:
Shimeng Yu, Arizona State University, USA
Doo Seok Jeong, Korea Institute of Science and Technology, South Korea

****Correspondence:***
Sapan Agarwal
sagarwa@sandia.gov;
James B. Aimone
jbaimon@sandia.gov

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *20 October 2015*
Accepted: *07 December 2015*
Published: *06 January 2016*

Citation:
Agarwal S, Quach T-T, Parekh O, Hsia AH, DeBenedictis EP, James CD, Marinella MJ and Aimone JB (2016) Energy Scaling Advantages of Resistive Memory Crossbar Based Computation and Its Application to Sparse Coding. Front. Neurosci. 9:484. doi: 10.3389/fnins.2015.00484

INTRODUCTION

As transistors start to approach fundamental physical limits and Moore's law slows down, new devices and architectures are needed to enable continued computing performance gains (Theis and Solomon, 2010). The computational ability of current microprocessors is limited by the power they consume. For data intensive applications, the computational energy is dominated by moving data between the processor, SRAM (static random access memory), and DRAM (dynamic random access memory). New approaches based on memristor or resistive memory (Chua, 1971; Waser and Aono, 2007; Strukov et al., 2008; Kim et al., 2012) crossbars can enable the processing of large amounts of data by significantly reducing data movement. One of the most promising applications for resistive memory crossbars is brain-inspired or neuromorphic computing (Jo et al., 2010; Ting et al., 2013; Hasan and Taha, 2014; Chen et al., 2015; Kim et al., 2015). The brain is perhaps the most energy-efficient computational system known, requiring only 1–100 femtoJoules per synaptic event (Merkle, 1989; Laughlin et al., 1998), efficiently solving complex problems such as pattern recognition on which conventional computers struggle. Consequently, there has been great interest in making

neuromorphic hardware (Cruz-Albrecht et al., 2013; Merolla et al., 2014). Resistive memories can effectively model some properties of neural synapses and the crossbar structure allows for high-density interconnectivity as found in the brain. For example, individual neurons in the cerebral cortex can receive roughly 10,000 input synapses from other neurons (Schüz and Palm, 1989).

Resistive memories are essentially programmable two terminal resistors. If a higher write voltage is applied to the device, the resistance will increase or decrease based on the sign of the voltage, allowing the resistance to be programmed. Consequently, it can be used to model a synapse. Its resistance acts like a weight that modulates the voltage applied to it. This has resulted in a large interest in developing neuromorphic systems based on it (Jo et al., 2010; Ting et al., 2013; Hasan and Taha, 2014; Kim et al., 2015). Each cell also has a very small area and the memory can be stacked in 3d when arranged in a crossbar structure. Therefore, industry is developing resistive memories to use as a digital replacement for flash memory (Jo et al., 2009; Chen, 2013; Chen et al., 2014; Cong et al., 2015).

A pressing question is whether neural-inspired computing systems are able to offer any resource advantage over more conventional digital computing systems. Neural-inspired systems are likely to take the form of a massively parallel collection of neuromorphic computing elements or cores that are each much simpler than conventional CPUs (Merolla et al., 2014). Conventionally, each neuromorphic core is based on a local SRAM memory array. This allows for data to be locally stored where it is used, eliminating the need to move large amounts of data. Simply organizing the computing system in this manner can provide 4–5 orders of magnitude reduction in computing energy (Cassidy et al., 2014). To get further benefits, the neuromorphic core should be based on an analog resistive memory crossbar array. Both digital and analog neuromorphic cores will have an execution-time advantage as parallelism is easier to leverage in a neuromorphic computational model where communication latency is drastically decreased. Nevertheless, in this work we avoid focusing on a new parallel architecture and instead focus on demonstrating a more fundamental advantage in energy.

We will show that performing certain computations on an analog resistive memory crossbar provides fundamental energy scaling advantages over a digital memory based implementation for finite precision computations. This is true for any architecture that uses a conventional digital memory array, even a digital resistive memory crossbar. In addition we give a concrete neural-inspired application, sparse coding, which can be implemented entirely in analog and reap the aforementioned energy advantage. A rich neural-inspired problem is sparse coding (Olshausen, 1996; Lee et al., 2008; Arora et al., 2015), where one seeks to use an overcomplete basis set to represent data with a sparse code. It is used in many applications including computer vision, object tracking, and more generally unsupervised learning. We will show that analog neural-inspired architectures are ideally suited for algorithms like sparse coding, and outline an implementation of a specific sparse coding algorithm.

Specifically, there are two key computational kernels that are more efficient on a crossbar. First, the crossbar can perform a parallel read or a vector-matrix multiplication as illustrated in **Figure 1**. Second, the crossbar can perform a parallel write or a rank-1 update where every weight is programmed based on the outer product of the row and column inputs. These two kernels form the basis of many neuromorphic algorithms.

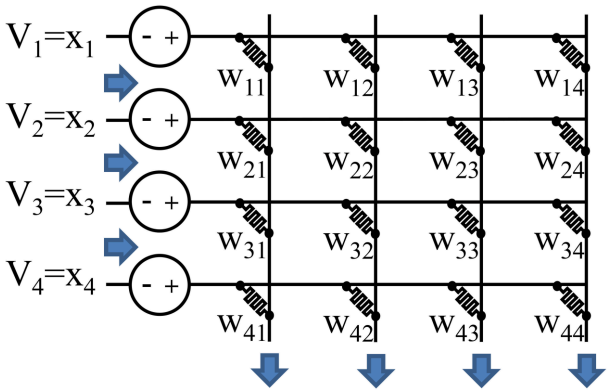


FIGURE 1 | Analog resistive memories can be used to reduce the energy of a vector-matrix multiply. The conductance of each resistive memory represents a weight. Analog input values are represented by the input voltages or input pulse lengths, and outputs are represented by current values. This allows all the read operations, multiplication operations, and sum operations to occur in a single step. A conventional architecture must perform these operations sequentially for each weight resulting in a higher energy and delay.

In this paper we analyze the energy required to perform a parallel read and show that for a fixed finite precision, the noise limited energy to compute a vector dot product can be independent of the size of the vector, $O(1)$, giving the analog resistive memory based dot product a significant scaling advantage over a digital approach. In the more likely situation of a capacitance limited energy, an $N \times N$ crossbar still has a factor of N scaling advantage over a digital memory. Similarly, writing a rank-1 update to a crossbar will also have a factor of N scaling advantage over a digital memory. We also analyze the energy cost of precision, energy scaling for communications, energy for accessing one row of the crossbar at a time and energy for accessing one element.

Next, we show that these computational kernels can be used with a sparse coding algorithm to make executing the algorithm $O(N)$ times more energy efficient.

RESULTS

Noise-Limited Parallel Read

A resistive memory crossbar can be used to perform a parallel analog vector-matrix multiplication as illustrated in **Figure 1**. Each column of the crossbar performs a vector dot product: $\sum_i x_i w_{ij}$ for column j. The inputs, x_i, are represented by either an analog voltage value or the length of a voltage pulse. The weights, w_{ij}, are represented by the resistive memory conductances. The multiplication is performed by leveraging $I = G \times V$, and the sum is performed by simply summing currents (or integrating the total current if the input, x_i, is encoded in the length of a pulse).

The absolute minimum energy to read the crossbar will be determined by the thermal noise in each resistor. For many computations we only need to know the result with some finite precision. Taking advantage of this allows the minimum energy to compute the vector dot product to be independent of the size of the vector, $O(1)$, when all the inputs and weights are positive.

To understand the tradeoff between precision and energy scaling, consider the minimum energy required to measure the current through N resistors with some signal to noise ratio (SNR). The signal strength we need to detect is dependent on the problem. If we want to keep the full precision of a digital computation, the minimum detectable signal must be proportional to the current through one resistor, I_o. On the other hand, in many computations we only need to know the final result to some precision. The minimum detectable signal for positive inputs/weights will be proportional to $N \times I_o$. This means that we are throwing away extra information and no longer want to detect the change in a single input, I_o. Effectively, we have a signal loss, α, of N, relative to a digital signal.

In many situations we will want negative weights or negative inputs. In this case the average signal might be zero. Nevertheless, the strength of the signal we want to detect will be given by the standard deviation of the signal. Consider inputs that have some distribution centered on zero, such as a Gaussian, and that have a variance proportional to I_o^2. The variance of N inputs will be proportional to $N \times I_o^2$. The strength of the signal we are detecting will be given by $\sqrt{N} \times I_o$ and the loss relative to digital is $\sqrt{N}$. Overall, the signal strength we want to detect is $\alpha \times I_o$ where α is between 1 and N.

The energy to read the resistors is given by:

$$\begin{aligned} \text{Energy} &= \text{Power per resistor} \times N \text{ resistors} \times \text{time} \\ &= V^2 G_o \times N \times \frac{1}{\Delta f} \end{aligned} \tag{1}$$

G_o is the conductance of each resistor and V is the voltage used to read the resistors. The operation speed, Δf, is determined by the thermal noise and the signal strength. We need to integrate the current for long enough to get the SNR we want. The thermal noise in N resistors is:

$$\text{Noise} = \left\langle \Delta I^2 \right\rangle = N \times \left(4k_b T \times G_o \times \Delta f\right) \tag{2}$$

The SNR is the signal strength divided by the noise:

$$SNR^2 = \frac{(\alpha I_o)^2}{\left\langle \Delta I^2 \right\rangle} = \frac{\alpha^2 \times I_o^2}{4k_b T \times N \times G_o \times \Delta f} \tag{3}$$

The current in a single resistor is given by $I_o = V \times G_o$. Using this and solving for time gives:

$$\frac{1}{\Delta f} = SNR^2 \times \frac{4k_b T \times N \times G_o}{\alpha^2 I_o^2} = SNR^2 \times \frac{N}{\alpha^2} \frac{4k_b T}{V^2 G_o} \tag{4}$$

Plugging this back into Equation (1) gives:

$$\begin{aligned} \text{Energy} &= V^2 G_o \times N \times SNR^2 \times \frac{N}{\alpha^2} \times \frac{4k_b T}{V^2 G_o} \\ &= 4k_b T \times \frac{N^2}{\alpha^2} \times SNR^2 \end{aligned} \tag{5}$$

For digital accuracy, $\alpha = 1$, and the vector dot product energy is $O(N^2)$ and is $O(N^3)$ for the full crossbar.

For finite output precision with positive inputs/weights, $\alpha = N$ and so the vector dot product energy is $O(1)$ and is $O(N)$ for the full crossbar. *Thus, the total noise limited dot product energy is the same regardless of the crossbar size.* As we increase the number of resistors and therefore signal strength, we can measure each device faster and with less precision and energy per device to get the same precision on the output. This is summarized in **Table 1**.

Capacitance-Limited Read

The previous analysis is only valid when the read energy is limited by the noise and not the capacitance. In particular, for fixed output precision with positive inputs/weights ($\alpha = N$), this is when Equation (5) is greater than the energy to charge the resistive memory and wire capacitance:

$$4k_b T \times SNR^2 > N \times C_{\text{per RRAM}} V^2 \tag{6}$$

If we assume we have a 1000×1000 crossbar, want a SNR of 100, and a resistive memory dominated capacitance of 18 aF (20×20 nm area, 5 nm thick capacitor with a relative permittivity, ε_r, of 25) we would need to perform the read at 100 mV or less to be noise limited. If a higher voltage is needed due to access devices or a larger crossbar is used, the energy will instead be capacitance limited.

For a capacitance-limited read energy, the crossbar will still be $O(N)$ times more energy efficient than an SRAM memory. The scaling advantage occurs because in a conventional SRAM memory, each row or wordline must be read or written one at a time. This means that the columns/bitlines and associated circuitry will need to be charged N times for N rows. In an analog crossbar, everything can be done in parallel and so the columns/bitlines and associated circuitry are only charged once. Thus, the crossbar is $O(N)$ times more energy efficient.

The most energy efficient way to organize a digital memory for performing vector-matrix multiplication is to have the matrix stored in an SRAM (or even a digital resistive memory) array. The energy increases by orders of magnitude if the weights are stored off chip. A typical SRAM cache is illustrated in **Figure 2**. To perform a vector-matrix multiply, at best we can read out one row/wordline at a time. For an $N \times N$ array, there will be N memory cells along each row/wordline. To read each memory cell along a row, we need to charge each bitline/column and

TABLE 1 | Energy scaling for different precision requirements.

	Minimum detectable signal	Loss relative to digital (α)	Full crossbar noise limited read energy
Digital Accuracy	I_O	1	$N \times 4k_b T \times N^2 \times SNR^2$
Fixed Output Precision	$\sqrt{N} \times I_O$	$\sqrt{N}$	$N \times 4k_b T \times N \times SNR^2$
Fixed Output Precision, only positive inputs/weights	$N \times I_O$	N	$N \times 4k_b T \times SNR^2$

run the read electronics/sense amp for each cell. Thus, the total energy is:

$$\begin{aligned}\text{Energy} &= N \text{ rows} \times N \text{ cells per row} \times E_{\text{digital bitline}} \\ &= N^2 \times E_{\text{digital bitline}} = O(N^3) \end{aligned} \quad (7)$$

The energy to charge each bitline, $E_{\text{digital bitline}}$, is proportional to the capacitance and therefore the length of the bitline: $E_{\text{digital bitline}} = NC_{\text{cell}}V^2$ where C_{cell} is the line capacitance across a single resistive memory cell. Thus, the energy scales as N^3.

In an analog resistive memory crossbar, all of the rows are charged in parallel and so the total energy is the sum of the energy to drive N rows and N columns:

$$\begin{aligned}\text{Energy} &= N \text{ rows} \times E_{\text{analog row}} + N \text{ columns} \times E_{\text{analog column}} \\ &= N \times (E_{\text{analog row}} + E_{\text{analog column}}) = O(N^2) \end{aligned} \quad (8)$$

The energy to charge each line also scales as the length of each line and therefore as N. Thus, the total energy for a crossbar scales as N^2 and is therefore is $O(N)$ times more energy efficient than an SRAM memory.

When engineering memory systems, there are a number of tricks that can be used to try to engineer around the scaling limits. If on-chip optical communications become feasible, the entire scaling tradeoff will be far better as the communication energy will effectively become independent of energy. Unfortunately, the energy and area overhead in converting from electrical to optical is currently orders of magnitude too high (Miller, 2009). 3d stacked memories will also scale better. In that case, this analysis would apply to a single layer of a 3d stacked memory. Digital memories can be broken into smaller subarrays with a processing unit near each sub-array. This is the principle behind processing in memory architectures. Nevertheless, even a minimal multiplier and adder logic block takes up a significant amount of area, limiting the minimum memory array size required to amortize the logic cost. If logic blocks are not placed next to each subarray, the bus capacitance to each sub array will cause the same scaling limits. Adiabatic computing can be used to tradeoff speed for the capacitance limited energy for both the digital and analog approaches.

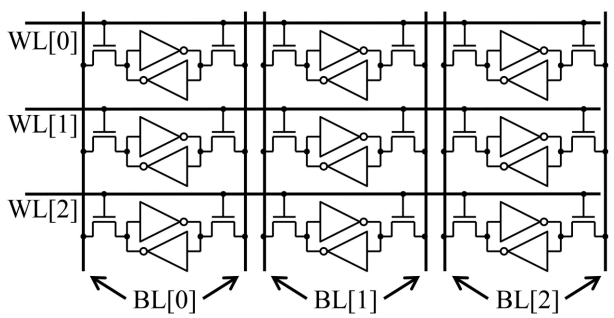


FIGURE 2 | A typical SRAM array. Each row/wordline must be accessed sequentially.

Parallel Write Energy

The energy scaling to write a SRAM cell will be identical to the energy to read the cell, Equation (7). N rows must each be written one at a time, and each row has N cells. When writing each cell, the energy to charge the bitline will be proportional to N. Consequently, the energy to write the array will scale as $O(N^3)$.

On an analog crossbar, we can perform a "parallel write" or a rank-1 update where every weight is programmed based on the outer product of the row and column inputs. An example of a parallel write is illustrated in **Figure 3**. The goal is to adjust the weight, W_{ij}, by the product of the inputs on the row, x_i, and column, y_j, of the weight:

$$W'_{ij} = W_{ij} + x_i \times y_j \quad (9)$$

An analog value for the row inputs, x_i, can be encoded by the length of the pulse. The longer the pulse the more the weight will change. The analog column inputs, y_j, can be encoded in the height of the pulse in order to achieve a multiplicative effect. The larger the voltage the more the weight will change for a given pulse duration. The exact write voltages will need to be adjusted to account for any non-linearities in the device. A parallel write can be done entirely in time as well (Kadetotad et al., 2015).

If the write is energy limited by the capacitance for the lines, the energy formula will be the same as in the read case and will be given by Equation (8). It will scale as $O(N^2)$ and is therefore is $O(N)$ times more energy efficient than an SRAM memory. However, each resistive memory will also typically require a fixed amount of current to program. If the energy is limited by the program current, the total energy will be given by number of resistive memories times the energy to program one:

$$E_{\text{write}} = N^2 I_{\text{write}} V_{\text{write}} \tau_{\text{write}} \quad (10)$$

I_{write} and V_{write} are the current and voltage, respectively, required to write a resistive memory. τ_{write} is the time required to write

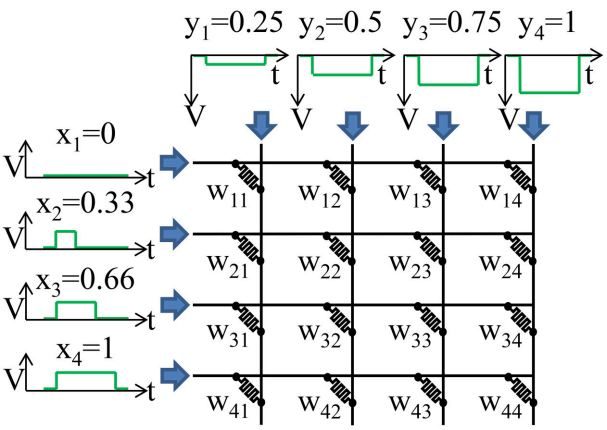


FIGURE 3 | A parallel write is illustrated. Weight W_{ij} is updated by $x_i \times y_j$. In order to achieve a multiplicative effect the x_i are encoded in time while the y_i are encoded in the height of a voltage pulse. The resistive memory will only train when x_i is non-zero. The height of y_i determines the strength of training when x_i is non-zero.

the resistive memory. In this case the energy still scales as $O(N^2)$ and so it still is $O(N)$ times more energy efficient than an SRAM memory.

If the write current or time is too large, it is possible that there will be a large constant factor that would make the energy scaling irrelevant. Fortunately, energies to fully write a resistive memory cell as low as 6 fJ have been demonstrated (Cheng et al., 2010). Furthermore, since we are operating the resistive memory as an analog memory with many levels, we do not want to fully write the cell. Rather, we only want to change the state by 1% or less, resulting in a corresponding reduction in the write energy per resistive memory. In this case the resistive memory energy will be on the same order of magnitude as the energy to charge the wires. (1% of 6 fJ is 60 aJ. The wire capacitance per resistive memory in a scaled technology node is likely to be on the order of 10's of attofarads [International Technology Roadmap for Semiconductors (ITRS, 2013)]. At 1V, that corresponds to 10's of attojoules as well).

Energy Cost of Precision

So far we have ignored the energy cost of computing at high precision. Analog crossbars are best at low to moderate precision as seen below. There are three values that can each have a different level of precision. Let the inputs, x_i, have a precisions in bits of b_{in}, the outputs have a precision of b_{out}, and the weights have a precision of b_w. Consider the noise-limited parallel read energy. The energy per column is given by Equation (5) and is proportional to the SNR^2 of the output. If we want $2^{b_{\text{out}}}$ levels on the output, the SNR must increase by $2^{b_{\text{out}}}$. This means that to create N outputs, at a precision of b_{out} bits, the crossbar energy will be on the order of the $O(N \times 2^{2b_{\text{out}}})$. If the crossbar is limited by the capacitance, the computation will already have sufficient precision and so the read/write energy will still be $O(N^2)$.

The thermal noise limited energy to process the output of the crossbar in analog at a certain precision will also scale as the voltage signal to noise ratio squared and therefore the number of output levels squared: $2^{2b_{\text{out}}}$ (Enz and Vittoz, 1996). If the output is converted from analog to digital, the D/A energy typically scales as the number of levels, $2^{b_{\text{out}}}$ (Murmann and Boser, 2007). Similarly, to convert a digital input to analog will scale as the number of input levels: $2^{b_{\text{in}}}$. Thus, we see that in the capacitance limited regime the total energy to read the crossbar is on the order of:

$$\text{Analog Capacitance Limited Energy} = O(N \times (N + 2^{2b_{\text{out}}} + 2^{b_{\text{in}}})) \quad (11)$$

or in the noise limited regime with positive inputs and weights it is:

$$\begin{aligned} \text{Analog Noise Limited Energy} &= O(N \times (2^{2b_{\text{out}}} + 2^{2b_{\text{out}}} + 2^{b_{\text{in}}})) \\ &= O(N \times (2^{2b_{\text{out}}} + 2^{b_{\text{in}}})) \end{aligned} \quad (12)$$

If we use a digital memory, we will need to store b_w bits for each weight. Consequently, we will need to multiply the energy by b_w: $E \sim O(N^3 b_w)$. We will also need to multiply each weight by its input and then sum the result. Assuming $b_w > b_{\text{in}}$, A single multiplication scales worse than $O(b_w \times \log(b_w))$ (Fürer, 2009) and so an entire crossbar with N^2 weights is at least $O(N^2 \times b_w \times \log(b_w))$. The sum operation will scale slower. Assuming $b_{\text{out}} < b_w \times \log(b_w)$ any neuron operations will also scale slower than the multiply operations. Thus, the digital energy is:

$$\text{Digital Energy} = O(N^2 \times b_w \times (N + \log(b_w))) \quad (13)$$

We see that for finite precision, analog is better, but if high floating point precision is required, digital is likely to be better.

Communications Energy

So far we have considered the energy of performing individual operations on a resistive memory crossbar. If we consider making a full system of multiple crossbars, the energy to communicate between crossbars can also be a significant component of the total energy. Consider the system shown in **Figure 4**. Each crossbar (or SRAM memory Merolla et al., 2014) is part of a neural core and each core communicates with the others over a communications bus. The energy to communicate between cores will be determined by the energy to charge the capacitance of the wire connecting two cores. Consequently the energy will be proportional to the capacitance and therefore the length of the wires. Assume that each core will communicate on average to a core that is fixed number of cores away. The size of each core will be determined by the size of the crossbar and so for an $N \times N$ crossbar, the length of an edge of a core will be of $O(N)$. Similarly, the length of wire to go a fixed number of cores away is of $O(N)$ and thus the energy is $O(N)$.

The key kernels discussed so far assume that a single operation drives an $N \times N$ matrix with N inputs and has N outputs. That means that each operation will have $O(N)$ communication events called spikes. Thus, we have $O(N)$ spikes and $O(N)$ energy cost per spike giving a total energy cost of $O(N^2)$. The energy to drive an SRAM based memory is of $O(N^3)$ and so the communications costs will be irrelevant for a large array. Indeed this is exactly the case in the IBM TrueNorth Architecture (Merolla et al., 2014). IBM projects that for an SRAM based system with a 256 × 256 core in a 10 nm technology the energy to communicate five

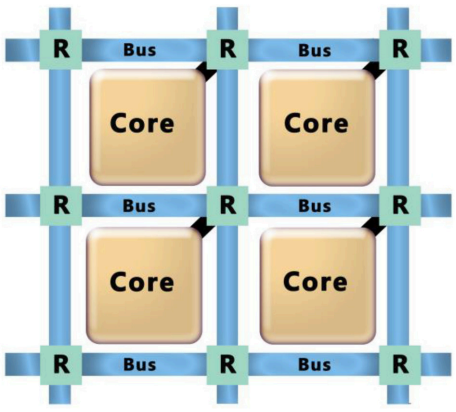


FIGURE 4 | A system would consist of individual crossbar based cores that communicate with each other through a communications bus and routing system.

cores away is 30 times lower than the energy to write the array. On the other hand, if we take advantage of an analog resistive memory crossbar, both the energy to read or write it and the energy to communicate will scale with $O(N^2)$. In this case, either the crossbar energy or the communication energy can dominate depending on the system architecture.

For algorithms that require cores that are far apart to communicate, the constant factor in the communication energy (the average communication distance) can be quite large and cause the communications energy to dominate. In this case, resistive memories can still provide a large constant factor reduction in the communication energy. Resistive memory potentially allows for terabytes of memory to be integrated onto a chip, while a chip using SRAM cannot hold more than 100 MB. This means that resistive memory can be $>10{,}000X$ denser than SRAM. Consequently, the edge length of a core can be reduced by $\sqrt{10{,}000} = 100X$. This would reduce the wire length and therefore communications energy by $100X$ or more. This is true regardless of whether the resistive memory is used as a digital or an analog memory.

Sparse Communications Algorithms

So far we have only considered kernels that operate on the entire $N \times N$ core at once. Some algorithms only operate on 1 row or even 1 element at a time. In these cases the energy scaling is very different.

First, consider an algorithm that operates on a single row at a time. Assume that in a given step a core receives an input, reads and writes one row and then sends out one communication spike to another core. We assume that on average the number of input spikes is the same as the number of output spikes so that the system remains stable (the spikes don't die off over time or blow up so that everything is spiking all the time). In this case, both the digital and the analog energy to read/write the crossbar scale as $O(N^2)$. This is because N bitlines need to be charged for one row and the energy per bitline scales as $O(N)$. Whether a digital or analog implementation is better will depend to the constant factors and exact system design. In both cases, using resistive memory for the memory reduces the wire lengths and therefore the power. The communications energy will scale as $O(N)$ for a single spike in/out since the core edge length scales as $O(N)$. This means the read/write energy will dominate as it scales as $O(N^2)$ and the communication scales as $O(N)$.

Next, consider an algorithm that operates on only a single element. In a given step, a core receives an input, reads a single memory element and sends a single output. In both the analog and digital cases we will charge one bitline and one wordline and so the energy will be proportional to the length of the line and will be $O(N)$. The communication energy for a single spike will also scale as $O(N)$, proportional to the core edge length. Both the communications energy and core energy need to be simultaneously optimized as they both scale as $O(N)$.

Rectangular Vs. Square Memory Arrays

So far we have assumed all our memory arrays are square $N \times N$ arrays. Let's consider an $N \times M$ array with N rows and M columns. For an analog resistive memory, the capacitance-limited read/write energy will still scale as the length of each column times the number of columns, $O(N \times M)$.

For a digital memory each row must be accessed sequentially and so the energy will scale as the number of rows times the length of each column times the number of columns: $O(N^2 \times M)$. If $N << M$, a digital rectangular array can be more efficient than a digital square array. Nevertheless, this is only true for a read if we only want output data along the M columns; i.e., we only perform the following multiplication, $\sum_i x_i w_{ij}$, where i represents the rows. If we also need to output data along the rows, i.e., perform the transpose operation: $\sum_j w_{ij} x_j$, the energy for that operation will scale as $O(N \times M^2)$, which would be worse than a square matrix.

In both cases, we have assumed that the data has the same shape, $N \times M$, as the memory. This allows us to perform the sum operation at the edge of each array and minimize the data movement. If the data is not the same shape as the array, the energy will be worse. Consider the situation shown in **Figure 5**. When the data is not the same shape as the array, we will need to move the data to a computational unit at a single location. The average wire length going to that unit (including both the wires in the array and outside of it) will be $O[\max(N,M)]$. Consequently, the energy will scale as the number of bits ($N \times$ M) times the total wire length $O[\max(N,M)]$ which is: $O[(N \times M) \times \max(N,M)]$. In this case a square array with an edge length of sqrt($N \times M$) would be the most efficient with an efficiency of $O(N^{3/2} \times M^{3/2})$. The same energy scaling applies to a write operation: the value to be written to the array depends on both row and column inputs and so it must be computed in one location and then communicated to the bitlines/columns in the array.

Sparse Coding Using a Resistive Memory Array

The energy efficiency of a resistive memory array can directly translate to making an algorithm more energy efficient. Consider

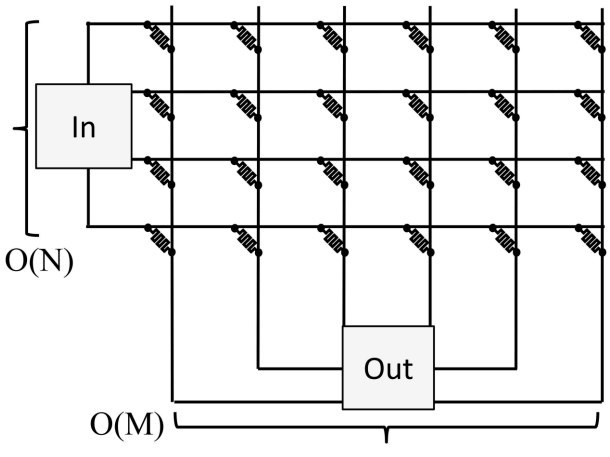


FIGURE 5 | If the data is not the same shape as the array, the input data will come from a single router, and the output data will need to go to a single computation unit. At best, the extra wire length to go to the input/output units plus the row/column wire length will be $O[\max(N,M)]$.

the problem of sparse coding. Sparse coding finds a set of basis vectors such that the linear combination of a few of these vectors is sufficient to explain each observation. Specifically, sparse coding finds matrix **A** that minimizes the following objective function (Olshausen, 1996; Lee et al., 2008):

$$\sum_{k=1}^{p} \left\| \mathbf{y}_k - \mathbf{A}\mathbf{x}_k \right\|^2 + \sum_{k=1}^{p} S(\mathbf{x}_k) \tag{14}$$

where **A** is an M by N matrix, $M<<N$, of basis vectors, p is the number of observations, $\mathbf{y}_k$, (of size M) is observation k, $\mathbf{x}_k$ (of size N) is the sparse representation of $\mathbf{y}_k$, and S is a sparsity cost such as the L_1 norm.

This problem is non-convex, but an approximate solution with guaranteed error bounds can be efficiently obtained via a recent algorithm by Arora et al. (2015) that extends the seminal gradient descent approach of Olshausen and Field (1997). In particular, we run t iterations or batches where we draw p samples and solve the following for each sample k:

$$\mathbf{x}_k = threshold_C \left(\mathbf{A}(t)^T \mathbf{y}_k \right) \tag{15}$$

where $\mathbf{A}(t)$ is the sparse coding matrix at iteration t. $threshold_C(.)$ is a thresholding function that keeps coordinates whose magnitude is at least C and zeros out the rest ensuring the code **x** is sparse. Next we compute a matrix update, $\Delta\mathbf{g}_k$, which is the outer-product of two vectors:

$$\Delta\mathbf{g}_k = \left(\mathbf{y}_k - \mathbf{A}(t)\mathbf{x}_k \right) \times \mathrm{sgn}(\mathbf{x}_k)^T \tag{16}$$

where sgn(.) is the sign function. All p updates need to be summed over a batch:

$$\mathbf{g}(t) = \sum_{k=1}^{p} \Delta\mathbf{g}_k \tag{17}$$

At the end of each batch, t, we update the sparse coding matrix:

$$\mathbf{A}(t+1) = \mathbf{A}(t) - \frac{\eta}{p}\mathbf{g}(t) \tag{18}$$

where η is the learning rate.

This sparse coding algorithm can be implemented efficiently with two resistive memory arrays. One array stores the sparse coding matrix $\mathbf{A}(t)$ while the second one stores the updates $\mathbf{\Delta g}_k$ during each batch. (Separate arrays should be used to minimize the wire length in each array) The arrays should be arranged to limit the wire length of the most frequent communication, $sgn(\mathbf{x}_k)$, to be of $O(M)$. This ensures that communications are not the limiting energy factor.

To analyze the energy efficiency, let's first consider all the operations performed for each sample k: For Equation (15), there are two operations being performed, the vector-matrix multiply, $\mathbf{A}(t)^T\mathbf{y}_k$, and the threshold function. In Equation (16), the resulting vector, $\mathbf{x}_k$, is multiplied by the matrix again, but without the transpose: $\mathbf{A}(t)\mathbf{x}_k$. Then a vector subtraction is performed: $\mathbf{y}_k - \mathbf{A}(t)\mathbf{x}_k$. Next, a sign operation $\mathrm{sgn}(\mathbf{x}_k)$ is performed. Finally, we have two vectors that need to be multiplied in an outer product and added to second matrix that stores the weight update. Moving data to the second matrix will incur a communications cost. After p samples, the summed updates in the second matrix, $\mathbf{g}(t)$, need to be copied, multiplied by η/p and written back to the original matrix, $\mathbf{A}(t)$. This operation can only operate a single row at a time as each weight needs to be read, communicated and written independently. This means analog will not have benefit over digital for this operation. Fortunately, it is only performed once every p samples. All the operations and their energy scaling are summarized in **Table 2**. In analog all the matrix operations will cost $O(N \times M)$. To maximize the digital energy efficiency, we assume we arrange a digital memory to be a square giving and energy cost of $O(N^{3/2} \times M^{3/2})$.

Let the inputs and outputs, have a precision in bits of b, and the weights have a precision of b_w. We consider finite precision such that $2^b < M$. This allows us to simplify Equation (11), the analog square matrix energy to be $E \sim O(N^2)$ and Equation (13) the digital energy to be $E \sim O(N^3)$. Here we are assuming that sparse coding algorithm will converge with a finite precision on the inputs and outputs. Neural-inspired algorithms like sparse coding tend to tolerate large amounts of noise, but the exact precision requirements should be studied for a practical implementation.

We can sum the energy scaling over all the operations listed in **Table 2**. Using the fact that $2^b<M<N<p$ (Arora et al., 2015) gives an overall analog energy scaling of: $O(N \times M \times p)$ and an overall digital energy scaling of $O(N^{3/2} \times M^{3/2} \times p)$. Thus, we see that analog has an overall energy advantage of $O[(N \times M)^{1/2}]$ or $O(N)$ if $N = M$.

TABLE 2 | The energy scaling for all the operations is given.

Operation	Analog energy scaling	Digital energy scaling	Repetitions per batch
MATRIX OPERATIONS			
Multiplication: $\mathbf{A}(t)^T \times \mathbf{y}_k$	$O(N \times M)$	$O(N^{3/2} \times M^{3/2})$	p
Multiplication: $\mathbf{A}(t) \times \mathbf{x}_k$	$O(N \times M)$	$O(N^{3/2} \times M^{3/2})$	p
Multiplication/Training: $(\mathbf{y}_k - \mathbf{A}(t)\mathbf{x}_k) \times \mathrm{sgn}(\mathbf{x}_k)^T$	$O(N \times M)$	$O(N^{3/2} \times M^{3/2})$	p
VECTOR OPERATIONS			
Threshold: $threshold_C \left(\mathbf{A}(t)^T\mathbf{y}_k\right)$	$O(N \times 2^b)$	$O(N \times b)$	p
Subtraction: $\mathbf{y}_k - \mathbf{A}(t)\mathbf{x}_k$	$O(M \times 2^b)$	$O(M \times b)$	p
Sign function: $\mathrm{sgn}(\mathbf{x}_k)$	$O(N \times 2^b)$	$O(N \times b)$	P
COMMUNICATION			
Vector: $(\mathbf{y}_k - \mathbf{A}(t)\mathbf{x}_k)$	$O(N \times M)$	$O(N^{1/2} \times M^{3/2})$	P
Vector: $\mathrm{sgn}(\mathbf{x}_k)^T$	$O(M \times N)$	$O(N^{3/2} \times M^{1/2})$	P
Matrix: $\mathbf{g}(t)$	$O(N^2 \times M)$	$O(N^{3/2} \times M^{3/2})$	1
SERIAL OPERATIONS			
Read: $\mathbf{g}(t)$	$O(N^2 \times M)$	$O(N^{3/2} \times M^{3/2})$	1
Write: $\mathbf{A}(t+1) = \mathbf{A}(t) - \frac{\eta}{p}\mathbf{g}(t)$	$O(N^2 \times M)$	$O(N^{3/2} \times M^{3/2})$	1

We consider the finite precision case such that $2^b < M$.

DISCUSSION

In this analysis we have deliberately avoided specifying constant factors as they can vary by orders of magnitude depending on the technology and design tradeoffs. Particular multiplicative constants apply only to today's hardware, but the big O remains whether new devices change these constants. For instance, the energy to write a resistive memory can be as low as 6 fJ (Cheng et al., 2010) or higher than 100 nJ (Mahalanabis et al., 2014). The energy for analog driving circuitry around a crossbar can also vary by orders of magnitude depending on the speed and circuit area tradeoffs. Depending on the algorithm, new semiconductor devices such as a spin based neuron (Sharad et al., 2014) could also drastically change the energy tradeoff.

Nevertheless, it is still useful to consider some specific numbers to understand what is plausible. In running an algorithm on a resistive memory array there are three key components to the energy, the parallel read energy, the parallel write energy and the energy for the driving circuitry. To find the capacitance limited read or write energy we need the capacitance per resistive memory element. The capacitance per element (wire + resistive memory) in an array for a 14 nm process as specified by ITRS will be around 50 aF. If we need to charge the wires to 1 V, that corresponds to 50 aJ per element. For an $N \times N$ array the total capacitance limited read or write energy would be 50 $\times N^2$ aJ. As discussed at the end of the Parallel Write Energy section, the current limited write energy could plausibly be on the same order of magnitude. The energy of the driving circuitry depends greatly on what computations are performed, but we can get an order of magnitude estimate by considering one of the most expensive analog operations, an analog to digital converter (ADC). ADCs that require only 0.85 fJ/level (or conversion step) have been demonstrated at 200 kHz (Tai et al., 2014). This means that for a 1000 $\times$ 1000 crossbar, the energy to run a six bit ADC is roughly the same as the energy to read/write a column of the crossbar. For higher precision ADCs, the ADC will dominate the energy, while for lower precision ADCs the crossbar will dominate the energy. In general, we see that the potential constant factors are on the same order of magnitude and consequently will be very technology dependent.

In order to understand the theoretical benefits of a crossbar, we have assumed ideal linear resistive memories. In practice there are many effects that can limit the performance of a resistive memory crossbar in a real algorithm. Access devices are required to be able to individually write a given resistive memory. This limits how low of a voltage can be used. Non-linearities in the resistive memories as well as those introduced by the access device mean that the amount a resistive memory writes will be dependent on its current state. Read and write noise limit the accuracy with which the resistive memories can be read or written. Parasitic voltage drops mean that devices far away from the drivers see a smaller voltage. Despite all of these effects, recent studies are indicating that iterative learning algorithms can tolerate and learn around moderate non-idealities (Burr et al., 2015; Chen et al., 2015; Cong et al., 2015). Given the potential energy scaling benefits of resistive memory crossbars, more work is need to design devices with fewer non-idealities and to better understand how various algorithms can perform given the non-idealities.

Overall, we have shown that the energy to perform a parallel read or parallel rank-1 write on an analog $N \times N$ resistive memory crossbar typically scales as $O(N^2)$ while a digital implementation scales as $O(N^3)$. Consequently, the analog crossbar has a scaling advantage of $O(N)$ in energy. The communications energy between neighboring crossbars scales as $O(N^2)$. Thus, communications are not as important for digital approaches, but once we take advantage of an analog approach the communication energy and computation energy are equally important. For algorithms that operate on only one row of a matrix at a time, both the digital and analog energy scales as $O(N^2)$ per row. Therefore, the better approach will depend on the specifics of a given system. Algorithms such as sparse coding can directly take advantage of the parallel write and parallel read to get an $O(N)$ energy savings.

Thus, we have shown that performing certain computations on an analog resistive memory crossbar provides fundamental energy scaling advantages over a conventional digital memory based implementation for low precision computations. This is true for any architecture that uses a conventional digital memory array, even a digital resistive memory crossbar. Fundamentally, a digital memory array must be accessed sequentially, one row at a time, while an entire analog memory crossbar can be accessed in parallel. Analog crossbars perform a multiply and accumulate at each crosspoint, while digital memories need to move the data to the edge of the array before it can be processed. In principle, a digital system could be organized to process data at every cell, but the area cost would become prohibitive. Alternatively, optimized digital neural systems will have a processing in memory (PIM; Gokhale et al., 1995) type architecture where simple operations are performed near a moderately sized memory array (Merolla et al., 2014). While this will give orders of magnitude reduction in energy compared to a CPU (Cassidy et al., 2014), the fundamental scaling advantages of an analog crossbar array can further reduce the energy by a few orders of magnitude.

AUTHOR CONTRIBUTIONS

SA, TQ, OP, ED, MM, CJ, and JA designed research; SA, TQ, OP performed research; AH provided technical guidance; and SA, TQ, OP wrote the paper.

ACKNOWLEDGMENTS

This work was supported by Sandia National Laboratories' Laboratory Directed Research and Development (LDRD) Program under the Hardware Acceleration of Adaptive Neural Algorithms Grand Challenge. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000

REFERENCES

Arora, S., Ge, R., Ma, T., and Moitra, A. (2015). Simple, efficient, and neural algorithms for sparse coding. *arXiv preprint arXiv:*1503.00778.

Burr, G. W., Shelby, R. M., Sidler, S., di Nolfo, C., Jang, J., Boybat, I., et al. (2015). Experimental demonstration and tolerancing of a large-scale neural network (165 000 Synapses) using phase-change memory as the synaptic weight element. *IEEE Trans. Electron Dev.* 62, 3498–3507. doi: 10.1109/TED.2015.2439635

Cassidy, A. S., Alvarez-Icaza, R., Akopyan, F., Sawada, J., Arthur, J. V., Merolla, P. A., et al. (2014). "Real-time scalable cortical computing at 46 giga-synaptic OPS/watt with ∼100× Speedup in Time-to-Solution and ∼100,000× reduction in energy-to-solution," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (New Orleans, LA: IEEE Press), 27–38.

Chen, A. (2013). A comprehensive crossbar array model with solutions for line resistance and nonlinear device characteristics. *IEEE Trans. Electron Dev.* 60, 1318–1326. doi: 10.1109/TED.2013.2246791

Chen, H.-Y., Gao, B., Li, H., Liu, R., Huang, P., Chen, Z., et al. (2014). "Towards high-speed, write-disturb tolerant 3D vertical RRAM arrays," in *Digest of Technical Papers, 2014 Symposium on*: IEEE *VLSI Technology (VLSI-Technology)* (Honolulu, HI), 1–2.

Chen, P.-Y., Kadetotad, D., Xu, Z., Mohanty, A., Lin, B., Ye, J., et al. (2015). "Technology-design co-optimization of resistive cross-point array for accelerating learning algorithms on chip," in *IEEE Design, Automation and Test in Europe (DATE) 2015* (Grenoble), 854–859.

Cheng, C. H., Tsai, C. Y., Chin, A., and Yeh, F. S. (2010). "High performance ultra-low energy RRAM with good retention and endurance," in *Electron Devices Meeting (IEDM), 2010 IEEE International* (San Francisco, CA), 19.14.11–19.14.14.

Chua, L. O. (1971). Memristor-The missing circuit element. *IEEE Trans. Circuit Theory* 18, 507–519. doi: 10.1109/TCT.1971.1083337

Cong, X., Dimin, N., Muralimanohar, N., Balasubramonian, R., Tao, Z., Shimeng, Y., et al. (2015). "Overcoming the challenges of crossbar resistive memory architectures," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)* (Burlingame, CA), 476–488.

Enz, C. C., and Vittoz, E. (1996). "CMOS low-power analog circuit design," in *Designing Low Power Digital Systems, Emerging Technologies (1996)* (Atlanta, GA: IEEE), 79–133.

Fürer, M. (2009). Faster Integer Multiplication. *SIAM J. Comput.* 39, 979–1005. doi: 10.1137/070711761

Gokhale, M., Holmes, B., and Iobst, K. (1995). Processing in memory: the Terasys massively parallel PIM array. *Computer* 28, 23–31. doi: 10.1109/2.375174

Hasan, R., and Taha, T. M. (2014). "Enabling back propagation training of memristor crossbar neuromorphic processors," in *Neural Networks (IJCNN), 2014 International Joint Conference on* (Beijing), 21–28.

International Technology Roadmap for Semiconductors (ITRS) (2013). *Edition* Available online at: http://www.itrs.net [Accessed].

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301. doi: 10.1021/nl904092h

Jo, S. H., Kim, K.-H., and Lu, W. (2009). High-density crossbar arrays based on a si memristive system. *Nano Lett.* 9, 870–874. doi: 10.1021/nl8037689

Cruz-Albrecht, J. M., Derosier, T., and Srinivasa, N. (2013). A scalable neural chip with synaptic electronics using CMOS integrated memristors. *Nanotechnology* 24:384011. doi: 10.1088/0957-4484/24/38/384011

Kadetotad, D., Zihan, X., Mohanty, A., Pai-Yu, C., Binbin, L., Jieping, Y., et al. (2015). Parallel Architecture with resistive crosspoint array for dictionary learning acceleration. *IEEE J. Emerg. Sel. Top. Circuits Syst.* 5, 194–204. doi: 10.1109/JETCAS.2015.2426495

Kim, K.-H., Gaba, S., Wheeler, D., Cruz-Albrecht, J. M., Hussain, T., Srinivasa, N., et al. (2012). A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications. *Nano Lett.* 12, 389–395. doi: 10.1021/nl203687n

Kim, Y., Zhang, Y., and Li, P. (2015). A reconfigurable digital neuromorphic processor with memristive synaptic crossbar for cognitive computing. *ACM J. Emerg. Technol. Comput. Syst.* 11, 38. doi: 10.1145/2700234

Laughlin, S. B., de Ruyter van Steveninck, R. R., and Anderson, J. C. (1998). The metabolic cost of neural information. *Nat. Neurosci.* 1, 36–41. doi: 10.1038/236

Lee, H., Ekanadham, C., and Ng, A. Y. (2008). "Sparse deep belief net model for visual area V2," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 873–880.

Mahalanabis, D., Barnaby, H. J., Gonzalez-Velo, Y., Kozicki, M. N., Vrudhula, S., and Dandamudi, P. (2014). Incremental resistance programming of programmable metallization cells for use as electronic synapses. *Solid State Electron.* 100, 39–44. doi: 10.1016/j.sse.2014.07.002

Merkle, R. C (1989). Energy limits to the computational power of the human brain. *Foresight Update 6*.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642

Miller, D. A. B. (2009). Device requirements for optical interconnects to silicon chips. *Proc. IEEE* 97, 1166–1185. doi: 10.1109/JPROC.2009.2014298

Murmann, B., and Boser, B. E. (2007). *Digitally Assisted Pipeline ADCs: Theory and Implementation*. Boston, MA: Springer Science & Business Media.

Olshausen, B. A. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381, 607–609. doi: 10.1038/381607a0

Olshausen, B. A., and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.* 37, 3311–3325. doi: 10.1016/S0042-6989(97)00169-7

Schüz, A., and Palm, G. (1989). Density of neurons and synapses in the cerebral cortex of the mouse. *J. Comp. Neurol.* 286, 442–455. doi: 10.1002/cne.902860404

Sharad, M., Deliang, F., Aitken, K., and Roy, K. (2014). Energy-efficient non-boolean computing with spin neurons and resistive memory. *Nanotechnol. IEEE Trans.* 13, 23–34. doi: 10.1109/TNANO.2013.2286424

Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. (2008). The missing memristor found. *Nature* 453, 80–83. doi: 10.1038/nature06932

Tai, H. Y., Hu, Y.-S., Chen, H.-W., and Chen, H.-S. (2014). "A 0.85fJ/conversion-step 10b 200kS/s subranging SAR ADC in 40nm CMOS," *IEEE International Solid-State Circuits Conference* (San Francisco, CA), 196–197,

Theis, T. N., and Solomon, P. M. (2010). In quest of the next switch: prospects for greatly reduced power dissipation in a successor to the silicon field-effect transistor. *Proc. IEEE* 98, 2005–2014. doi: 10.1109/JPROC.2010.2066531

Ting, C., Yuchao, Y., and Wei, L. (2013). Building neuromorphic circuits with memristive devices. *IEEE Circuits Syst. Mag.* 13, 56–73. doi: 10.1109/MCAS.2013.2256260

Waser, R., and Aono, M. (2007). Nanoionics-based resistive switching memories. *Nat. Mater.* 6, 833–840. doi: 10.1038/nmat2023

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 24 December 2015
doi: 10.3389/fnins.2015.00488

Modeling and Experimental Demonstration of a Hopfield Network Analog-to-Digital Converter with Hybrid CMOS/Memristor Circuits

Xinjie Guo[1†], Farnood Merrikh-Bayat[1†], Ligang Gao[1], Brian D. Hoskins[1], Fabien Alibart[2], Bernabe Linares-Barranco[3], Luke Theogarajan[1], Christof Teuscher[4] and Dmitri B. Strukov[1]*

[1] Department of Electrical and Computer Engineering, University of California, Santa Barbara, Santa Barbara, CA, USA, [2] Centre National de la Recherche Scientifique, Lille, France, [3] Instituto de Microelectronica de Sevilla (Consejo Superior de Investigaciones Científicas and University of Seville), Seville, Spain, [4] Department of Electrical and Computer Engineering, Portland State University, Portland, OR, USA

OPEN ACCESS

Edited by:
Emmanuel Michael Drakakis, Imperial College London, UK

Reviewed by:
Duygu Kuzum, University of California, San Diego, USA
Shimeng Yu, Arizona State University, USA

***Correspondence:**
Dmitri B. Strukov
strukov@ece.ucsb.edu

† These authors have contributed equally to this work.

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *20 October 2015*
Accepted: *07 December 2015*
Published: *24 December 2015*

Citation:
Guo X, Merrikh-Bayat F, Gao L, Hoskins BD, Alibart F, Linares-Barranco B, Theogarajan L, Teuscher C and Strukov DB (2015) Modeling and Experimental Demonstration of a Hopfield Network Analog-to-Digital Converter with Hybrid CMOS/Memristor Circuits. Front. Neurosci. 9:488. doi: 10.3389/fnins.2015.00488

The purpose of this work was to demonstrate the feasibility of building recurrent artificial neural networks with hybrid complementary metal oxide semiconductor (CMOS)/memristor circuits. To do so, we modeled a Hopfield network implementing an analog-to-digital converter (ADC) with up to 8 bits of precision. Major shortcomings affecting the ADC's precision, such as the non-ideal behavior of CMOS circuitry and the specific limitations of memristors, were investigated and an effective solution was proposed, capitalizing on the in-field programmability of memristors. The theoretical work was validated experimentally by demonstrating the successful operation of a 4-bit ADC circuit implemented with discrete Pt/TiO_{2-x}/Pt memristors and CMOS integrated circuit components.

Keywords: Hopfield network, recurrent neural network, hybrid circuits, memristor, resistive switching, analog-to-digital conversion

INTRODUCTION

Recurrent artificial neural networks are an important computational paradigm capable of solving a number of optimization problems (Hopfield, 1984; Tank and Hopfield, 1986). One classic example of such networks is a Hopfield analog-to-digital converter (Tank and Hopfield, 1986; Lee and Sheu, 1989; Smith and Portmann, 1989). Although such a circuit may be of little practical use, and inferior, for example, to similar-style feed forward-type ADC implementations (Chigusa and Tanaka, 1990), it belongs to a broader constrained optimization class of networks which minimize certain pre-programmed energy functions and have several applications in control and signal processing (Tank and Hopfield, 1986). The Hopfield network ADC circuit also represents an important bridge between computational neuroscience and circuit design, and an understanding of the potential shortcomings of such a relatively simple circuit is therefore important for implementing more complex recurrent neural networks.

An example of a 4-bit Hopfield network ADC is shown in **Figure 1** (Tank and Hopfield, 1986). The originally proposed network consists of an array of linear resistors (also called *weights* or *synapses*) and four peripheral inverting amplifiers (*neurons*). Each neuron receives currents from the input and reference lines and from all other neurons via corresponding synapses. The analog input voltage V_S is converted to the digital code $V_3V_2V_1V_0$, i.e.,

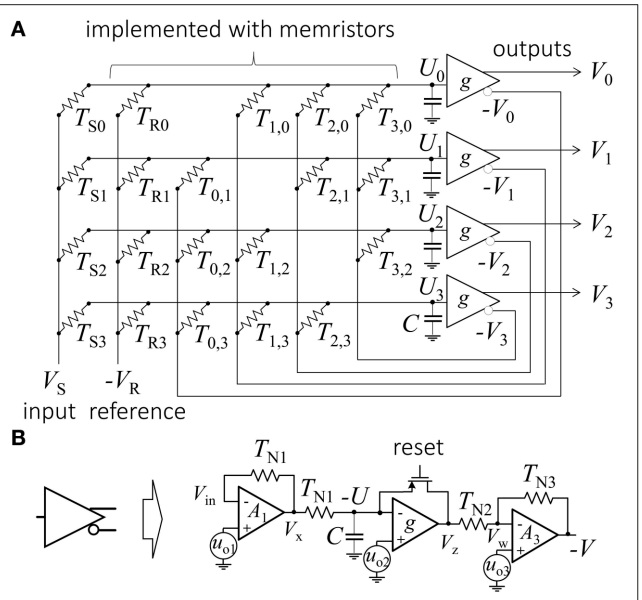


FIGURE 1 | (A) Conventional Hopfield network implementation of a 4-bit ADC and **(B)** specific implementation of a neuron as considered in this paper.

$$V_S = \sum_{i=0}^{3} 2^i V_i \tag{1}$$

by first forcing all neuron outputs to zero (Lee and Sheu, 1989) and then letting the system evolve to the appropriate stationary state.

To understand how the Hopfield network performs the ADC operation, let us first describe its electrical behavior. Assuming leakage-free neurons with infinite input and zero output impedances, the dynamic equation governing the system evolution of the input voltage U_j of the j-th neuron is described as:

$$C\dot{U}_j = -\sum_i T_{ij} V_i - T_j U_j + I_j \tag{2a}$$

$$V_i = g(U_i), \tag{2b}$$

where $g(\cdot)$ is a neuron activation function, C is the neuron's input capacitance, T_{ij} is a conductance of the synapse connecting the output of the i-th neuron with the input of the j-th neuron, while

$$I_j = T_{Sj} V_S - T_{Rj} V_R, \tag{3}$$

$$T_j = T_{Sj} + T_{Rj} + \Sigma_i T_{ij} \tag{4}$$

are the corresponding effective offset input current and effective input conductance for the j-th neuron. Here V_R is a reference voltage, while T_R and T_S are conductances of reference and input weights, respectively (**Figure 1A**). Note that neuron input U_i can be either positive or negative, but the output of the neuron is either zero or positive. The inverted outputs of the neurons, which are fed back to the network, are therefore either negative or zero. One activation function suitable for such mapping is the sigmoid function $1/(1+\exp[-U])$. Neuron output needs to be inverted to keep the feedback weights positive and thus to allow physical implementation with passive devices, such as resistors[1].

Alternatively, the Hopfield network operation can be described by an energy function. The evolution of the dynamic system described by Equation (2) is equivalent to a minimization of the energy function:

$$E = \frac{1}{2}\sum_{ij} T_{ij} V_i V_j - \sum_j V_j I_j - \sum_j T_j \int_0^{V_j} g^{-1}(V)\, dV \tag{5}$$

where the last term can be neglected for very steep transfer functions (Hopfield, 1984). In Tank and Hopfield (1986), showed that a 4-bit ADC task (Equation 1) can be described by the following energy function:

$$E = \frac{1}{2}(V_S - \sum_{i=0}^{3} 2^i V_i)^2 - \frac{1}{2}\sum_{i=0}^{3} 2^{2i} V_i (V_i - 1) \tag{6}$$

Here the first term tends to satisfy Equation (1), while the second tends to force each digital output V_i to be either "0" or "1." After rearranging the terms in Equation (6) and comparing the result with Equation(5), the appropriate weights for performing the ADC task are:

$$T_{ij} = 2^{(i+j)},\ T_{Sj} = 2^j, T_{Rj} = 2^{(2j-1)}. \tag{7}$$

In the Hopfield ADC network, the number of synapses grows quadratically with the number of neurons. Compact implementation of the synapses is therefore required if such circuits are to be practical. This is certainly challenging to achieve with conventional CMOS technology, because, according to Equation (7), it requires analog weights with a relatively large dynamic range, i.e., in the order of 2^{2N}, where N is the bit precision. Weights can be stored digitally, but this approach comes with a large overhead (Moopenn et al., 1990). On the other hand, analog CMOS implementations of the synapses have to cope with the mismatch issues often encountered in CMOS circuits (Indeveri et al., 2011). Consequently, several attempts have been made to implement synapses with alternative, nonconventional technologies. In some of the early implementations of Hopfield networks, weights were realized as corresponding thin film (Jackel et al., 1987) or metal line (Graf et al., 1986; Schwartz et al., 1987) conductance values, patterned using e-beam lithography and reactive-ion-etching. The main limitation of these approaches was that the weights were essentially one-time programmable, with rather crude accuracy. A much more attractive solution was very recently demonstrated in Eryilmaz et al. (2014), which describes a Hopfield network implementation with synapses based on phase change memory paired with conventional field-effect transistors. That work, together with other recent advances in device

[1]The sign of the first term on the left in Equation (2a), and of all right hand terms in Equation (5), is different from that of the original paper (Hopfield, 1984). In this work we assume that all weights are strictly positive, making it necessary explicitly to flip the neuron feedback signal sign.

technologies (Wu et al., 2012; Zhang et al., 2012) revived interest in the theoretical modeling of recurrent neural networks based on hybrid circuits (Waser et al., 2009; Strukov and Kohlstedt, 2012; Lehtonen et al., 2014; Rakkiyappan et al., 2014; Walls and Likharev, 2014).

This paper explores the implementation of synapses with an emerging, very promising type of memory devices, namely metal-oxide resistive switching devices ("memristor") (Wu et al., 2012; Zhang et al., 2012). In the next section we discuss the general implementation details of the Hopfield network ADC, including the memristor devices which were utilized in the experimental setup. This is followed by a theoretical analysis of the considered hybrid circuits' sensitivity to certain representative sources of non-ideal behavior and discussion of a possible solution to such problems. The theoretical results were validated with SPICE simulations (Section Simulation Results) and experimental work (Section Experimental Results). The paper concludes with a Discussion section. It should be noted that preliminary experimental results, without any theoretical analysis, were reported earlier in Gao et al. (2013a), where we first presented a Hopfield network implementation with metal-oxide memristors. The only other relevant experimental work on memristor-based Hopfield networks that we are aware of was published recently in Hu et al. (2015). However, the network demonstrated in Hu et al. (2015) was based on 9 memristors whereas the circuit presented in this work involves 16.

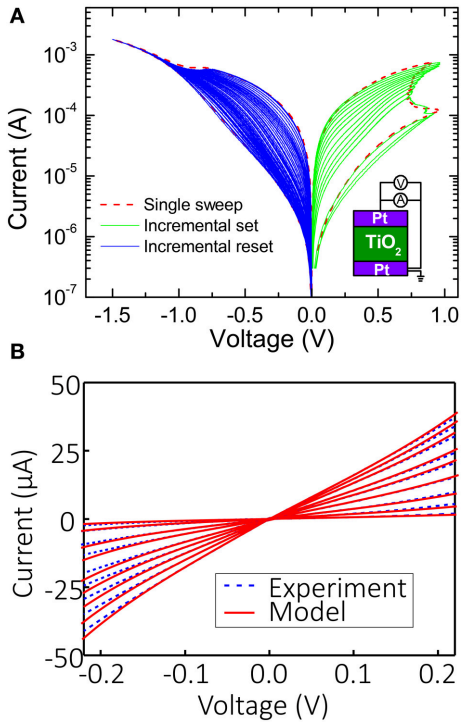


FIGURE 2 | (A) Typical *I-V* curves with current-controlled set and voltage-controlled reset switching for the considered $Pt/TiO_{2-x}/Pt$ memristors. **(B)** Modeling of static *I-V* curves at small disturb-free voltages for several different states. The fitting parameters are $\beta = 1$, $\alpha_1 = 14.7\,V^{-3}$, $\alpha_2 = -5.9 \times 10^4\ \Omega V^{-3}$, $\alpha_3 = 1.5 \times 10^8\ \Omega^2 V^{-3}$ for $V > 0$, and $\alpha_1 = 34.6\,V^{-3}$, $\alpha_2 = -1.9 \times 10^5\ \Omega V^{-3}$, $\alpha_3 = 3.65 \times 10^8\ \Omega^2 V^{-3}$ for $V < 0$.

MATERIALS AND METHODS FOR HOPFIELD NETWORK IMPLEMENTATION WITH HYBRID CIRCUITS

Following on from our earlier works (Alibart et al., 2013; Gao et al., 2013b; Merrikh-Bayat et al., 2014), we here consider the implementation of a hybrid CMOS/memristive circuit (**Figure 1**). In this circuit, density-critical synapses are implemented with $Pt/TiO_{2-x}/Pt$ memristive devices, while neurons are implemented by CMOS circuits.

In their simplest form, memristors are two-terminal passive elements, the conductance of which can be modulated reversibly by applying electrical stress. Due to the simple structure and ionic nature of their memory mechanism, metal-oxide memristors have excellent scaling prospects, often combined with fast, low energy switching and high retention (Strukov and Kohlstedt, 2012). Many metal oxide based memristors can also be switched continuously, i.e., in analog manner, by applying electrical bias (current or voltage pulses) with gradually increasing amplitude and/or duration.

Figure 2A shows typical continuous switching *I-V*s for the considered $Pt/TiO_{2-x}/Pt$ devices (Alibart et al., 2012). The devices were implemented in "bone-structure" geometry with an active area of $\sim 1\,\mu m^2$ using the atomic layer deposition technique. An evaporated Ti/Pt bottom electrode (5 nm/25 nm) was patterned by conventional optical lithography on a Si/SiO_2 substrate (500 μm/200 nm, respectively). A 30 nm TiO_2 switching layer was then realized by atomic layer deposition at 200°C using Titanium Isopropoxide ($C_{12}H_{28}O_4Ti$) and water as precursor and reactant, respectively. A Pt/Au electrode (15 nm/25 nm) was evaporated on top of the TiO_2 blanket layer, and the device was finally rapidly annealed at 500°C in an N_2 and N_2+O_2 atmosphere for 5 min to improve the crystallinity of the TiO_2 material. Details of the fabrication and characterization of the considered memristors are given in Alibart et al. (2012).

After programming the memristors to the desired resistance, it was important for their state to remain unchanged during operation of the Hopfield network, so to prevent any disturbance the voltage drop across them was always kept within the $|V| \leq 0.2$ V "disturb-free" range (Alibart et al., 2012).

The static *I-V* characteristics (i.e., those within disturb-free regime) for several different memory states are shown in **Figure 2B**. To assist SPICE simulation, the experimental *I-V* curves at small biases were fitted by the following static equation with a single memory state *G*:

$$I = GV + \beta(\alpha_1 G + \alpha_2 G^2 + \alpha_3 G^3)V^4. \quad (8)$$

The need to keep the voltage drop across memristive devices small also affects neuron design. A simple leaky operational amplifier (op-amp) integrator could be sufficient to implement neuron functionality, but ensuring disturb-free operation with such a design is not easy. This issue was resolved by implementing neurons with three op-amps connected in series (**Figure 1B**). The

first op-amp was an inverting amplifier which held virtual ground even if the neuron's output was saturated. The second op-amp was an open loop amplifier implementing a sign-like activation function. The field effect transistor in the negative feedback of this op-amp was initially turned on to force the neuron's outputs to zero (i.e., to set into initial state before computing output) and then turned off during network convergence. The last op-amp inverted the signal and ensured that the neuron output was within the $-0.2\,\mathrm{V} \le V \le 0$ voltage range. Note that since the neuron bandwidth was mainly determined by the input capacitance of the second amplifier, and the other sources of parasitic capacitance could be neglected for simplicity, the capacitive load of the second amplifier (**Figure 1B**) was effectively a neuron input capacitance (**Figure 1A**).

Assuming ideal op-amps and no possibility of saturation by the first and last amplifiers, the dynamic equation for this neuron design can be written as:

$$C\dot{U}_j = -\sum_i T_{ij}V_i - T_{N1}U_j + I_j \tag{9a}$$

$$V_j = -T_{N2}/T_{N3}g(U_j), \tag{9b}$$

where $g()$ is a transfer function of the second op-amp (see Appendix for more details on derivation).

For a very steep transfer function, the second term in the right hand part of Equation (9a) can be neglected (Hopfield, 1984). The network is then described by the original energy function (Equation 5) and the weights are proportional to those defined in Equation (7), i.e.,

$$T_{ij}' = 5T_{ij},\ T_{Sj}' = T_{Sj},\ T_{Rj}' = 5T_{Rj}, \tag{10}$$

where the additional coefficient 5 is due to the reduced, i.e., 0.2 V, output voltage corresponding to digital "1" in the considered circuit [as opposed to output voltage 1 V assumed in the original ADC energy function in Equation (6) for ADC and the weights in Equation (7) derived from that energy function].

The physical implementation of this Hopfield network ADC posed a number of additional challenges. However, it should first be mentioned that variations in neuron delay and input capacitances, which may result in oscillatory behavior and the settling in of false energy minima (Lee and Sheu, 1989; Smith and Portmann, 1989), were not a problem in our case thanks to the slow operating speed, which was enforced to reduce capacitive coupling. The specific problems regarding the considered implementation were offsets in virtual ground, resulting from the voltage offsets (u_o) and limited gain (A) of the op-amps (**Figure 1B**). Another, somewhat less severe, problem was the nonlinear conductance of the memristive devices (defined via parameter β–, see Equation 8). In the Appendix it is shown how limited gain and non-zero offset result in an additional constant term I_0 in dynamical equation (Equation A7), which can be factored into the reference weights as follows:

$$T_{Rj}'' = T_{Rj}' + I_{0j}/V_R. \tag{11}$$

The Hopfield network with practical, non-ideal neurons can still therefore be approximated by the original energy equation and it should be possible to circumvent the effects of limited gain and voltage offset by fine-tuning the reference weights. This idea was verified via SPICE modeling and experimental work, as described in the next section.

RESULTS

Simulation Results

Using Equation (8) for the memristors and SPICE models for the IC components, in the next series of simulations we studied how particular non-ideal behavior affects differential (DNL) and integral (INL) nonlinearities in ADC circuits (van de Plassche, 2003). **Figure 3A** shows INL and DNL as a function of the open loop DC gain, which was varied simultaneously for all three op-amps, assuming ideal memristors with $\beta = 0$ and no voltage offset. Note that in this simulation, the gain-bandwidth product (*GBP*) was increased proportionally to the open loop DC gain, and was equal to 3 MHz at $A_{DC} = 2 \times 10^5$. Because the circuit operated at about 1.5 KHz, the effective gain $A \approx A_{DC}/100$ for all simulations (and also for the experimental work discussed below). **Figure 3B** shows the impact of the

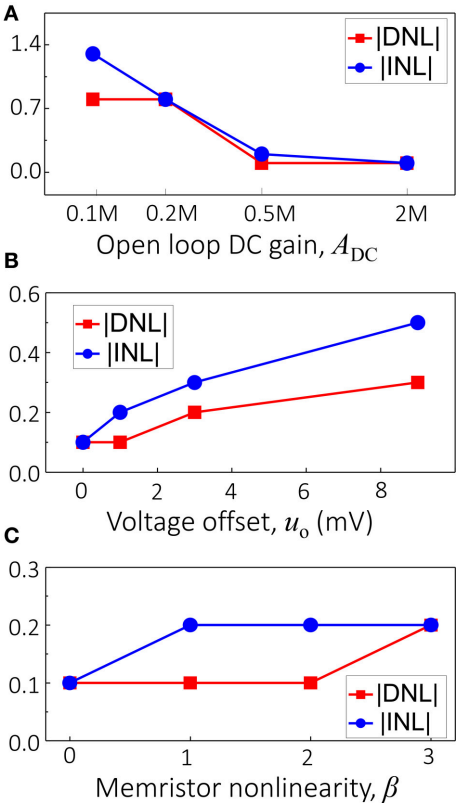


FIGURE 3 | Theoretical analysis of the performance sensitivity of a 4-bit Hopfield network ADC with respect to (A) open-loop DC gain, (B) voltage offsets in the operational amplifiers, and (C) the nonlinearity of memristive devices.

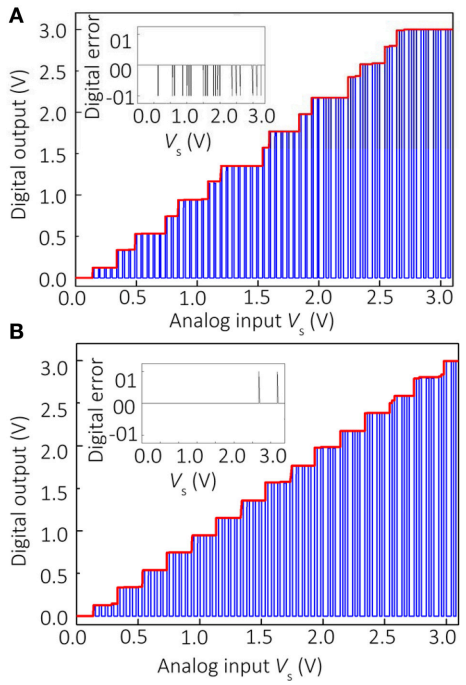


FIGURE 4 | Simulation results for (A) the original and (B) the optimized 4-bit Hopfield network ADC with $\beta = 1$, $A_{DC} = 2 \times 10^5$, and $u_o = 3$ mV voltage offset, which are representative parameters for the experimental setup. For the optimized network, $T_R" = 0.97\ T_{R1}$, $T_{R2}" = 0.86\ T_{R2}$, $T_{R3}" = 0.95\ T_{R3}$, $T_{R4}" = 0.97\ T_{R4}$.

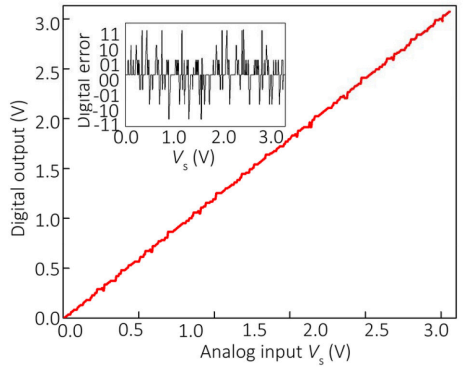


FIGURE 5 | Simulation results for the optimized 8-bit Hopfield network ADC with $T_{R1}" = 0.8\ T_{R1}$, $T_{R2}" = 0.81\ T_{R2}$, $T_{R3}" = 0.89\ T_{R3}$, $T_{R4}" = 0.83\ T_{R4}$, $T_{R5}" = 0.55\ T_{R5}$, $T_{R6}" = 0.74\ T_{R6}$, $T_{R7}" = 0.71\ T_{R7}$, $T_{R8}" = 0.75\ T_{R8}$. All other parameters are equal to those used for **Figure 4**.

voltage offset on DNL and INL (simulated as an offset on the ground nodes), which was varied simultaneously for all three op-amps. Finally, **Figure 3C** shows the effect of *I-V* nonlinearity, which was varied by changing constant β in Equation (8), assuming all other parameters of the network to be close to ideal, i.e., that the voltage offset $u_o = 0$ and the open loop DC gain $A_{DC} = 10^6$. Note that for $\beta > 0$, the memristor weights were chosen in such a way that the conductance of the device at −0.2V matched the corresponding values prescribed by Equation (10).

The results shown in **Figure 3** confirm the significant individual contribution of the considered sources of non-ideal behavior on the ADC's performance. **Figure 4A** shows the simulation results considering all these factors together for the specific values $u_o = 3$ mV, $\beta = 1$, $A_{DC} = 2 \times 10^5$, and $GBP = 3$ MHz, which are representative of the experimental setup. The gain and voltage offset values were taken from the specifications of the discrete IC op-amps used in the experiment. Clearly, the ADC output is distorted and contains numerous errors, with the largest contribution to INL being due to finite gain (**Figure 3**). **Figures 4B**, **5** show the simulation results with new values for the reference weights calculated according to Equation (11) for the 4-bit and 8-bit ADCs, respectively. The results shown in these figures confirm that non-ideal behavior in op-amps, such as limited gain and voltage offsets, can be efficiently compensated by fine-tuning memristors.

Experimental Results

The simulation results were also validated experimentally by implementing a 4-bit Hopfield network ADC in a breadboard setup consisting of Pt/TiO_{2-x}/Pt memristive devices and discrete IC CMOS components (**Figure 6A**). The memristor chips were assembled in standard 40-pin DIP packages by wire-bonding 20 standalone memristive devices. Because input voltage range is $0 \le V_S \le V_S^{max} = 3.0$ V, the weights T_S were realized with regular resistors[2]. The discrete memristors and other IC components were then connected as shown in **Figure 1** with external wires.

The memristors implementing feedback and reference weights were first tuned ex-situ using a previously developed algorithm (Alibart et al., 2012) to the values defined by Equation (10). The ex-situ tuning for each memristor was performed individually before the devices were connected in a circuit. This was done to simplify the experiment and it is worth mentioning that in general, it should be possible to tune memristors after they are connected in the crossbar circuit, as it was experimentally demonstrated by our group for standalone devices connected in crossbar circuits (Alibart et al., 2013; Gao et al., 2013c) and integrated passive crossbar circuits (Prezioso et al., 2015a,b).

As was discussed in Sections Materials and Methods for Hopfield Network Implementation with Hybrid Circuits and Results, limited gain and voltage offsets of operational amplifiers can be compensated by adjusting reference weights according to Equations (11, A12). To demonstrate in-field configurability of memristors, the reference weights were fine-tuned in-situ. In particular, reference weights were adjusted to ensure correct outputs at four particular input voltages, when V_S is equal to 1/16, 1/8, 1/4, and 1/2 of its maximum value. The tuning is performed first for $V_S = 1/16\ V_S^{max}$, for which the correct operation of ADC assumes that the least significant output bit V_0 flips from 0 to 1 (corresponding to voltage 0.2 V in our case), which is ensured

[2]In principal, input voltage range could be decreased by increasing input weights correspondingly. However, such rescaling would require larger a dynamic range of conductances to implement (Equation 6), and this was not possible with the considered memristive devices.

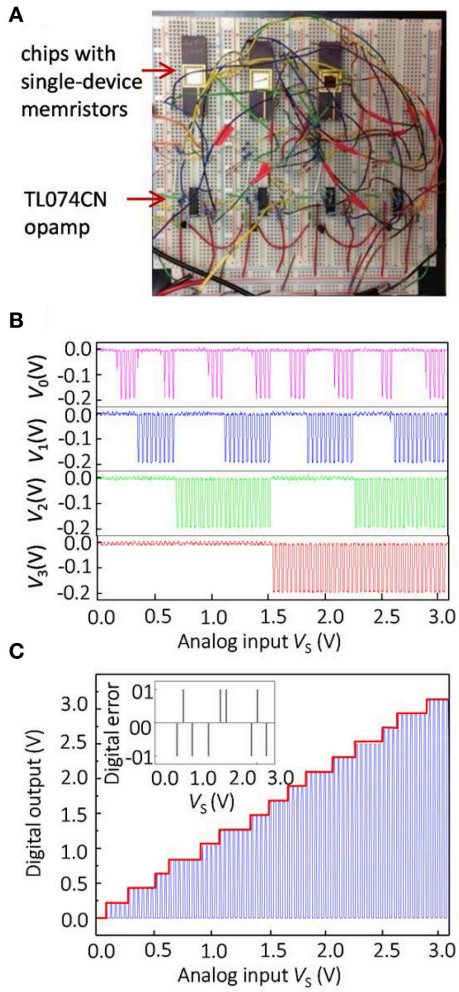


FIGURE 6 | Experimental results for the optimized 4-bit Hopfield ADC: (A) experimental setup, (B) measured outputs for every output channel, and (C) measured transfer characteristics.

by fine-tuning reference weight T_{R0}. Similarly, the output bit V_1 should flip from 0 to 1 when $V_S = 1/8\ V_S^{max}$, which is ensured by fine-tuning reference weight T_{R1} and so on. Because we started fine-tuning from the least significant output, it is sufficient to fine-tune only one corresponding reference weight at a time for a particular input voltage, which greatly simplified in-situ tuning procedure. Also, the direction of adjustment was always straightforward to determine due to monotonic dependence of the input voltage at which a particular output bit flips from 0 to 1, on the corresponding reference weight (Equation 11).

DISCUSSION

The network parameters for the experimental work are summarized in **Table 1**. Although there were a few A/D conversion errors in the experimental work (**Figure 6**), the results are comparable with the simulations of the optimized network, and much better than those obtained for the unoptimized network. The experimental results for the unoptimized network were significantly worse in comparison with the simulation, and are not shown in this paper.

TABLE 1 | Parameters for the experimentally demonstrated Hopfield network ADC.

Feed-back	Conductance (S@0.2V)	Reference	Conductance (S@0.2V)
$T2,1$	2e-5	T_{1R}	4.75e-6
$T3,1$	4e-5	T_{2R}	2.19e-5
$T4,1$	7.9e-5	T_{3R}	9.33e-5
$T1,2$	2e-5	T_{4R}	41.85e-5
$T3,2$	7.9e-5	Input	Conductance (S)
$T4,2$	15e-5	T_{1S}	8.33e-6
$T1,3$	4e-5	T_{2S}	1.67e-5
$T2,3$	7.9e-5	T_{3S}	3.33e-5
$T4,3$	30.9e-5	T_{4S}	6.67e-5
$T1,4$	7.9e-5	Neuron	Conductance (S)
$T2,4$	15e-5	T_{N1}	1e-3
$T3,4$	30.9e-5	T_{N2}	1e-5
		T_{N3}	5e-4

It is worth mentioning that for the considered memristors drift of conductive state over time was negligible due to highly nonlinear switching kinetics specific to these devices (Alibart et al., 2012, 2013; Prezioso et al., 2015a). In principle, for other types of memristors with inferior retention properties it should be possible to occasionally fine-tune memristor state to cope with conductance drift. A related issue might be measurement noise upon reading the state of the memristor, e.g., due to the fluctuations in the device conductance over time, which is sometimes observed as random telegraph noise (Gao et al., 2012, 2013b; Prezioso et al., 2015b). Such noise can be tolerated by performing quasi DC read measurements, however, the downside would be potentially much slower tuning process.

To conclude, in this work we investigated hybrid CMOS/metal-oxide-memristor circuit implementation of a Hopfield recurrent neural network performing analog-to-digital conversion tasks. We showed that naïve implementation of such networks, with weights prescribed by the original theory, produces many conversion errors, mainly due to the non-ideal behavior of the CMOS components in the integrated circuit. We then proposed a method of adjusting weights in the Hopfield network to overcome the non-ideal behavior of the network components and successfully validated this technique experimentally on a 4-bit ADC circuit. The ability to fine-tune the conductances of memristors in a circuit was essential for implementing the proposed technique. In our opinion, the work carried out proved to be an important milestone and its results will be valuable for implementing more practical large-scale recurrent neural networks with CMOS/memristor circuits. Experimental research into CMOS/memristor neural networks is still very scarce and, to the best of our knowledge, the demonstrated Hopfield network is the most complex network of its type reported to date. From a broader perspective, this paper demonstrates one of the main advantages of utilizing memristors in analog circuits, namely the feasibility of

fine-tuning memristors after fabrication to overcome variations in analog circuits.

AUTHOR CONTRIBUTIONS

XG and FMB performed simulation work. FMB, XG, and LG performed the experimental demo. LG, BH, and FA fabricated devices. DS supervised the project. All discussed the results.

ACKNOWLEDGMENTS

This work was supported by NSF grant CCF-1028378 and by the Air Force Office of Scientific Research (AFOSR) under MURI grant FA9550-12-1-0038, and by Spanish grant TEC2012-37868-C04-01 (BIOSENSE) (with support from the European Regional Development Fund).

REFERENCES

Alibart, F., Gao, L., Hoskins, B. D., and Strukov, D. B. (2012). High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. *Nanotechnology* 23:075201. doi: 10.1088/0957-4484/23/7/075201

Alibart, F., Zamanidoost, E., and Strukov, D. B. (2013). Pattern classification by memristive crossbar circuits using ex situ and in situ training. *Nat. Commun.* 4, 2072. doi: 10.1038/ncomms3072

Chigusa, Y., and Tanaka, M. (1990). "A neural-like feed-forward ADC," in *Proc. ISCAS'90* (New Orleans, LA), 2959–2962.

Eryilmaz, S. B., Kuzum, D., Jeyasingh, R., Kim, S., BrightSky, M., Lam, C., et al. (2014). Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array. *Front. Neurosci.* 8:205. doi: 10.3389/fnins.2014.00205

Gao, L., Merrikh-Bayat, F., Alibart, F., Guo, X., Hoskins, B. D., Cheng, K.-T., et al. (2013a). "Digital-to-analog and analog-to-digital conversion with metal oxide memristors for ultra-low power computing," in *Proc. NanoArch'13* (New York, NY), 19–22.

Gao, L., Alibart, F., and Strukov, D. (2013b). Programmable CMOS/memristor threshold logic. *IEEE Trans. Nanotechnol.* 12, 115–119. doi: 10.1109/TNANO.2013.2241075

Gao, L., Alibart, F., and Strukov, D., B. (2013c). "A high resolution nonvolatile analog memory ionic devices," in *Proc. Non-Volatile Memories Workshop* (San Diego, CA).

Gao, L., Alibart, F., and Strukov, D. B. (2012). Analog-input analog-weight dot-product operation with Ag/a-Si/Pt memristive devices," in *Proc. VLSI-SoC'12* (Santa Cruz, CA), 88–93.

Graf, H. P., Jackel, L. D., Howard, R. E., Straughn, B., Denker, J. S., Hubbard, W. et al. (1986). VLSI implemetnation of a nerual network memory with several hundreds of neurons. *AIP Conf. Proc.* 151, 182–187. doi: 10.1063/1.36253

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neruons. *Proc. Natl. Acad. Sci. U.S.A.* 81, 3088–3092. doi: 10.1073/pnas.81.10.3088

Hu, S. G., Liu, Y., Liu, Z., Chen, T. P., Wang, J. J., Yu, Q., et al. (2015). Associative memory realized by a configurable memristive Hopfiled neural network. *Nat. Commun.* 6, 7522. doi: 10.1038/ncomms8522

Indiveri, G., Linares-Barranco, B., Hamilton, T. J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5:73. doi: 10.3389/fnins.2011.00073

Jackel, L. D., Graf, H. P., and Howard, R. E. (1987). Electronic neural network chips. *Appl. Opt.* 26, 5077–5080. doi: 10.1364/AO.26.005077

Lee, B. W., and Sheu, B. J. (1989). Design of a neural-based A/D converter using modified Hopfield network. *IEEE J. Solid-State Circ.* 24, 1129–1135. doi: 10.1109/4.34101

Lehtonen, E., Poikonen, J. H., Laiho, M., and Kanerva, P. (2014). Large-scale memristive associative memories. *IEEE Trans. VLSI* 22, 562–574. doi: 10.1109/TVLSI.2013.2250319

Merrikh-Bayat, F., Alibart, F., Gao, L., and Strukov, D. B. (2014). "A reconfigurable FIR filter with memristor-based weights," in: *Proc. ISCAS'15* (Melbourne, VIC).

Moopenn, A., Duong, T., and Thakoor, A. P. (1990). . *Adv. Neural Inofmra Digital-analog hybrid synapse chips for electronic nerual networks t. Proces. Syst.* 2, 769–776.

Prezioso, M., Kataeva, I., Merrikh-Bayat, F., Hoskins, B., Adam, G., Sota, T., et al. (2015b). "Modeling and implementation of firing-rate neuromorphic-network classifiers with bilayer Pt/Al_2O_3/TiO_{2-x}/Pt memristors," in *IEDM'15*. (Washington, DC).

Prezioso, M., Merrikh Bayat, F., Hoskins, B. D., Adam, G. C., Likharev, K. K., and Strukov, D. B. (2015a). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521, 61–64. doi: 10.1038/nature14441

Rakkiyappan, R., Chandrasekar, A., Laksmanan, S., and Ju Park, H. (2014). State estimation of memristor-based recurrent neural networks with time-varying delays based on passivity theory. *Complexity* 19, 32–43. doi: 10.1002/cplx.21482

Schwartz, D. B., Howard, R. E., Denker, J. S., Epworth, R. W., Graf, H. P., Hubbard, W., et al. (1987). Dynamics of microfabriacted eletronic neural networks. *Appl. Phys. Lett.* 50, 1110–1112.

Smith, M. J. S., and Portmann, C. L. (1989). Practical design and analysis of a simple "neural" optimization circuit. *IEEE Trans. Circ. Syst.* 36, 42–50. doi: 10.1109/31.16562

Strukov, D. B., and Kohlstedt, H. (2012). Resistive switching phenomena in thin films: Materials, devices, and applications. *MRS Bull.* 37, 108–114. doi: 10.1557/mrs.2012.2

Tank, D. W., and Hopfield, J. J. (1986). Simple neural optimization networks—an A/D converter, signal decision circuit, and a linear-programming circuit. *IEEE Trans. Circ. Syst.* 33, 533–541. doi: 10.1109/TCS.1986.1085953

van de Plassche, R. J. (2003). *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters, 2nd Edn.* Norwell, MA: Kluwer Academic Publishers.

Walls, T. J., and Likharev, K. K. (2014). Self-organization in autonomous, recurrent, firing-rate CrossNets with quasi-hebbian plasticity. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 819–824. doi: 10.1109/TNNLS.2013.2280904

Waser, R., Dittmann, R., Staikov, G., and Szot, K. (2009). Redox—based resistive switching memories–nanoionic mechanisms, prospects, and challenges. *Adv. Mat.* 21, 2632–2663. doi: 10.1002/adma.200900375

Wu, A., Wen, S., and Zeng, Z. (2012). Synchronization control of a class of memristor-based recurrent neural networks. *Inform. Sci.* 183, 106–116. doi: 10.1016/j.ins.2011.07.044

Zhang, G., Shen, Y., and Sun, J. (2012). Global exponential stability of a class of memristor-based recurrent neural networks with time-varying delays. *Neurocomputing* 97, 149–154. doi: 10.1016/j.neucom.2012.05.002

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX

Assuming negligible op-amp input currents and output impedances, the Hopfield network is described by the following equations, which also account for limited gain and voltage offsets:

$$V_{xj} = A_{1j}\left(u_{o1j} - V_{inj}\right), \tag{A1}$$

$$V_{zj} = g(u_{o2j} + U_j), \tag{A2}$$

$$-V_j = A_{3j}\left(u_{o3j} - V_{wj}\right), \tag{A3}$$

$$T_{\mathrm{N1}}\left(V_{inj} - V_{xj}\right) = T_{\mathrm{R}j}\left(-V_{\mathrm{R}} - V_{inj}\right) + T_{\mathrm{S}j}\left(V_{\mathrm{S}} - V_{inj}\right) + \sum_i T_{ij}(-V_i - V_{inj}) \tag{A4}$$

$$-C\dot{U}_j = T_{\mathrm{N1}}(V_{xj} + U_j), \tag{A5}$$

$$T_{N2}\left(V_{zj} - V_{\mathrm{w}j}\right) = T_{\mathrm{N3}}\left(V_{\mathrm{w}j} + V_j\right). \tag{A6}$$

Solving these equations results in the following dynamic equation

$$a_j C\dot{U'}_j = -\sum_i T_{ij}V'_i - a_j T_{\mathrm{N1}}U'_j + I_j + I_{oj} \tag{A7a}$$

$$b_j V'_j = g\left(U'_j\right), \tag{A7b}$$

where $g()$ is a transfer function of the saturating amplifier implemented with the second op-amp, and

$$U'_j = u_{o2j} + U, \tag{A8}$$

$$V'_i = u_{o3j}(1 + T_{\mathrm{N3}j}/T_{\mathrm{N2}j})/b_j + V_i, \tag{A9}$$

$$a_j = 1 + (1 + T_j/T_{\mathrm{N1}j})/A_{1j}, \tag{A10}$$

$$b_j = T_{\mathrm{N3}j}/T_{\mathrm{N2}j} + (1 + T_{\mathrm{N3}j}/T_{\mathrm{N2}j})/A_{3j}, \tag{A11}$$

$$I_{oj} = -\left(T_{\mathrm{N1}j} + T_j\right)u_{o1j} + a_j T_{\mathrm{N1}j}u_{o2j} + \frac{1 + \frac{T_{\mathrm{N3}j}}{T_{\mathrm{N2}j}}}{b_j}\sum_i T_{ij}u_{o3j} \tag{A12}$$

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 02 October 2015
doi: 10.3389/fnins.2015.00357

Implementation of a spike-based perceptron learning rule using TiO_{2-x} memristors

Hesham Mostafa[1]*, *Ali Khiat*[2], *Alexander Serb*[2], *Christian G. Mayr*[1], *Giacomo Indiveri*[1] and *Themis Prodromakis*[2]

[1] Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland, [2] Nanoelectronics and Nanotechnology Research Group, School of Electronics and Computer Science, University of Southampton, UK

Synaptic plasticity plays a crucial role in allowing neural networks to learn and adapt to various input environments. Neuromorphic systems need to implement plastic synapses to obtain basic "cognitive" capabilities such as learning. One promising and scalable approach for implementing neuromorphic synapses is to use nano-scale memristors as synaptic elements. In this paper we propose a hybrid CMOS-memristor system comprising CMOS neurons interconnected through TiO_{2-x} memristors, and spike-based learning circuits that modulate the conductance of the memristive synapse elements according to a spike-based Perceptron plasticity rule. We highlight a number of advantages for using this spike-based plasticity rule as compared to other forms of spike timing dependent plasticity (STDP) rules. We provide experimental proof-of-concept results with two silicon neurons connected through a memristive synapse that show how the CMOS plasticity circuits can induce stable changes in memristor conductances, giving rise to increased synaptic strength after a potentiation episode and to decreased strength after a depression episode.

Keywords: synaptic plasticity, silicon neurons, memristors, neuromorphic architectures, learning, perceptron

OPEN ACCESS

Edited by:
Gert Cauwenberghs, University Of California, San Diego, USA

Reviewed by:
Siddharth Joshi, University of California, San Diego, USA
Duygu Kuzum, University of California, San Diego, USA
Shahar Kvatinsky, Technion - Israel Institute of Technology, Israel

***Correspondence:**
Hesham Mostafa, Institute of Neuroinformatics, University of Zurich and ETH Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland
hesham@ini.uzh.ch

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *08 June 2015*
Accepted: *18 September 2015*
Published: *02 October 2015*

Citation:
Mostafa H, Khiat A, Serb A, Mayr CG, Indiveri G and Prodromakis T (2015) Implementation of a spike-based perceptron learning rule using TiO_{2-x} memristors. Front. Neurosci. 9:357. doi: 10.3389/fnins.2015.00357

1. Introduction

Biological networks provide a tantalizing proof of the existence of a physically implementable computing architecture that is distributed, fault-tolerant, adaptive, and that outperforms conventional architectures in many important problems such as visual processing and motor control. This has motivated the development of various neuromorphic computing systems whose architectures reflect the general organizational principles of nervous systems in an effort to partially reproduce the immense efficiency advantage that biological computation exhibits in some problems. These neuromorphic systems are organized as populations of excitatory and inhibitory spiking neurons with configurable synaptic connections (FACETS, 2005–2009; Navaridas et al., 2013; Benjamin et al., 2014; Merolla et al., 2014; Ning et al., 2015).

Synapses outnumber neurons by several orders of magnitude in biological neural networks (Binzegger et al., 2004). Reproducing these biological features in neuromorphic electronic circuits presents a scaling problem, as integrating thousands of dedicated synapse circuits per neuron can quickly become infeasible for systems that require a large number of neurons (Schemmel et al., 2007). This scaling problem has traditionally been solved by either treating synapses as simple linear elements and time-multiplexing spikes from many pre-synaptic

sources onto the same linear circuit (Benjamin et al., 2014), or by treating them as basic binary elements that can be set either *'on'* or *'off'* externally, without learning abilities (Merolla et al., 2014).

Real synapses, however, exhibit non-linear phenomena like spike timing dependent plasticity (STDP) that modulate the weight of an individual synapse based on the activity of the pre- and post-synaptic neurons (Bi and Poo, 1998). The modulation of synaptic weights through plasticity has been shown to greatly increase the range of computations that neural networks can perform (Abbott and Regehr, 2004). Capturing the plasticity properties of real synapses in analog neuromorphic hardware requires the use of distinct physical circuits/elements for each synapse. In conventional CMOS, this can lead to restrictions on scalability. Some potential solutions to the scalability issues in pure CMOS technology involve the use of very large integrated structures (e.g., up to a full wafer, Schemmel et al., 2012) or the adoption of deep submicron technologies (Noack et al., 2015). Scalability restrictions however can be greatly relaxed if one resorts to compact nano-scale circuit elements that can reproduce the plasticity properties of real synapses.

One potential candidate for these elements is the "memristor." Chua (1971) described the memristor as an element which *behaves somewhat like a non-linear resistor with memory*. Since HP first linked resistively switching devices with the concept of a memristor (Strukov et al., 2008), work on memristive devices has mostly focused on digital storage and logic functions (Linn et al., 2012; You et al., 2014), but there are also applications as analog/multi-level storage (Moreno et al., 2010; Shuai et al., 2013) and even memristive encryption (Lin and Wang, 2010; Du et al., 2014). In the neuromorphic community, memristors are seen as ideal devices for synapse implementations, as they combine three key functions in one device. Memristors can implement biologically realistic synaptic weight updates, i.e., learning (Jo et al., 2010), they can carry out long term multi-valued weight storage, and they can also communicate weighted pre-synaptic activity to the postsynaptic side (Saighi et al., 2015), significantly relaxing scalability restrictions (Indiveri et al., 2013).

Typically, plasticity in these memristive synapses is evoked by applying specific waveforms to the two terminals of the memristor, with the waveforms aligned to pre- respectively postsynaptic pulses (Jo et al., 2010). The correlation of the waveforms across the memristor in turn implements STDP-like plasticity (Mayr et al., 2012), with the form of the STDP curve defined by the applied wave shape (Serrano-Gotarredona et al., 2013). Both hardware and software models of plasticity based on the basic STDP mechanism are typically chosen primarily for their simplicity (Mayr and Partzsch, 2010). It has been argued however that more elaborate models of plasticity are required to reproduce the experimental evidence obtained from more complex synaptic plasticity experiments in real neural systems, and to implement algorithms that can learn to store and classify correlated patterns (Senn and Fusi, 2005; Sjöström et al., 2008; Lisman and Spruston, 2010).

In this work we present a neuromorphic implementation of one of these extended plasticity models that implements a spike-based Perceptron learning algorithm (Brader et al., 2007), which makes use of both analog CMOS circuits and TiO_{2-x} memristive devices. Compared to the more widely used STDP paradigm, the implementation of this learning algorithm on memristors does not employ the postsynaptic spike timing. Instead, it relies on the correlation of presynaptic spikes with signals derived from the postsynaptic neuron, such as its membrane potential and a measurement of its recent spiking activity. These requirements lead to a novel and quite different approach to the CMOS driver circuits which does not require the generation of temporally long waveforms on the pre- or postsynaptic sides.

In addition to spike timing, plasticity in biological synapses also depends on the firing rate of the post-synaptic neuron (Sjöström et al., 2001), a phenomenon that can not be captured by pair-wise STDP mechanisms (Pfister et al., 2006). The spike-based perceptron learning rule explicitly contains a term that reflects the recent firing rate of the neuron and is thus able to realize the rate-dependence of synaptic weight updates. The rule is also able to realize weight updates that depend on pre-post spike timing even though it does not explicitly depend on the post-synaptic spike times. Instead, it uses the membrane potential of the post-synaptic neuron as an indirect estimator of post-synaptic firing times. The rule is thus able to reasonably match the behavior of biological synapses while having a functional form that can be implemented efficiently on pure CMOS or on hybrid CMOS-memristor neuromorphic systems.

We introduce the spike-based Perceptron learning model in Section 2.1 and the TiO_{2-x} memristive devices employed in this implementation in Section 2.2. The adaptation of the learning model to memristors is described in Section 2.3. Considerations for crossbar operation of this paradigm are given in Section 2.4. Section 3.1 shows basic results characterizing operation of the memristors. Characterization of the learning CMOS driver circuits implemented in VLSI are detailed in Section 3.2. Finally, results from implementing the spike-based Perceptron learning with the CMOS driver circuits on the memristors are presented in Section 3.3.

2. Materials and Methods

2.1. The Plasticity Model

The spike-based Perceptron learning model of long-term plasticity has been introduced in Brader et al. (2007) based on earlier work in Fusi et al. (2000). The model represents a synapse with two stable weights, potentiated and depressed, whereby the transition between the two stable weights is done in an analog or graded manner. The synaptic weight $X(t)$ is influenced by a combination of pre- and post-synaptic activity, namely the pre-synaptic spike time t_{pre} and the value of the post-synaptic neuron membrane voltage $V_{mem}(t)$ and intra-cellular calcium concentration $C(t)$. A pre-synaptic spike arriving at t_{pre} reads the instantaneous post-synaptic values $V_{mem}(t_{pre})$ and $C(t_{pre})$. The change in $X(t)$ depends on these instantaneous values in the following way:

$$\mathrm{X} \rightarrow \mathrm{X}+a \quad if \quad \{V_{mem}(t_{pre}) > \theta_V \quad and \quad \theta^l_{up} < C(t_{pre}) < \theta^h_{up}\} \tag{1}$$

$$\mathrm{X} \rightarrow \mathrm{X}-b \quad if \quad \{V_{mem}(t_{pre}) \leq \theta_V \quad and \quad \theta^l_{down} < C(t_{pre}) < \theta^h_{down}\}, \tag{2}$$

where a and b are jump sizes and θ_V is a voltage threshold. In other words, $X(t)$ is increased if $V_{mem}(t)$ is elevated (above θ_V) when the pre-synaptic spike arrives and decreased when $V_{mem}(t)$ is low at time t_{pre} provided that the calcium variable $C(t)$ is in the correct range. θ^l_{up}, θ^h_{up}, θ^l_{down}, and θ^h_{down} are thresholds on the calcium variable. The calcium variable $C(t)$ is an auxiliary variable that is a low-pass filtered version of the post-synaptic spikes (see Brader et al., 2007, for details). The variable $C(t)$ is incremented by J_C at each post-synaptic spike time t_i, where J_C reflects the magnitude of spike-triggered calcium influx into the cell. $C(t)$ decays with a time constant τ_C:

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

The dependence of the weight updates on $C(t)$ allows the learning rule to enable/disable the weight updates based on the long-term average of post-synaptic activity. $X(t)$ continuously drifts toward one of two stable values based on whether it is above or below the threshold θ_X:

$$\frac{d\mathrm{X}}{dt} = \alpha \quad if \quad \mathrm{X} > \theta_X \tag{4}$$

$$\frac{d\mathrm{X}}{dt} = -\beta \quad if \quad \mathrm{X} \leq \theta_X \tag{5}$$

The weight $X(t)$ is bounded above and below by the two stable states X_{high} and X_{low} which are not shown in the equations to simplify the notation. **Figure 1** illustrates the relevant waveforms and parameters of the spike-based Perceptron learning rule.

The dynamics of the membrane potential variable, V_{mem}, which is used in Equations (1) and (2), depend on the neuron model used. The original neuron model used with the perceptron-learning rule is the simple constant leak integrate and fire neuron model (Brader et al., 2007). The neuron circuit we have in our neuromorphic chip, however, implements the more realistic adaptive exponential integrate and fire neuron model. This neuron circuit and the underlying model are described in detail in Indiveri et al. (2011) and Ning et al. (2015). The interaction between this adaptive exponential integrate and fire silicon neuron and the spike-based perceptron-learning rule is described in Indiveri et al. (2010).

Although the spike-based plasticity rule described above has been shown to reproduce, on average, the classical STDP phenomenology (Brader et al., 2007), it differs from the vast majority of spike-timing plasticity rules in that it does not explicitly depend on the precise timing of both pre- and post-synaptic neuron spikes. The compatibility with the classical STDP learning rule comes about through the rule's dependence on the post-synaptic neuron's membrane potential: a pre-synaptic spike that occurs when the post-synaptic membrane potential is high will potentiate the synapse and will likely produce a post-synaptic spike shortly after. Thus, the synapse tends to get potentiated in pre-before-post scenarios. The synapse also tends to get depressed in post-before-pre scenarios because the membrane potential is usually low for a few milliseconds after a post-synaptic spike is emitted, and a pre-synaptic spike arriving in this interval will depress the synapse.

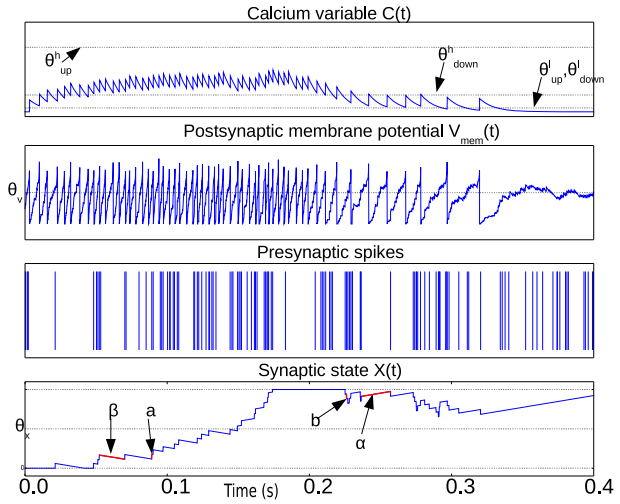


FIGURE 1 | Illustration waveforms of the spike-based perceptron learning rule showing key parameters from Equations (1–5). The Calcium variable plot shows the ranges defined by $\theta^l_{up}, \theta^h_{up}, \theta^l_{down}, \theta^h_{down}$ within which synaptic plasticity is active according to Equations (1) and (2). The post-synaptic neuron membrane potential plot shows the threshold θ_V. Incoming synapses can be depressed (potentiated) if $V_{mem}(t)$ is below (above) θ_V. The bottom plot showing the synaptic state $X(t)$ illustrates the jump and drift mechanism. On each pre-synaptic spike, the mutually exclusive conditions in Equations (1) and (2) are evaluated. If the condition in Equations (1) and (2) is fulfilled, the synaptic state jumps up (down) by a step a (b). The synaptic state is continuously drifting to a high or low state depending on whether it is above or below the threshold θ_X, respectively.

The spike-based Perceptron plasticity rule also has access to post-synaptic neuron's rate information through the $C(t)$ signal. This allows it to reproduce effects beyond classical pair-wise STDP such as increased potentiation at high post-synaptic firing rates and increased depression at low post-synaptic firing rates (Sjöström et al., 2001). These effects arise in more complicated STDP models such as triplet STDP (Pfister et al., 2006; Mayr and Partzsch, 2010). The absence of explicit dependence on the post-synaptic neuron's firing times thus does not diminish the biological plausibility or the computational power of the spike-based Perceptron learning rule.

For the purpose of pure CMOS VLSI implementation (Chicca et al., 2014), this plasticity model is interesting because it can learn a graded response to an input pattern but on long time scales, the weight $X(t)$ drifts to one of two stable states and is thus easy to store long-term. In the hybrid CMOS-memristor architecture that we propose in this paper, however, the weight drift (Equations 4 and 5) is not implemented. The memristor conductance (weight) only changes on pre-synaptic spikes.

Weight drift or the bi-stable synaptic dynamics of the perceptron learning rule can be useful in consolidating the synaptic changes and making the synaptic weight more robust against spurious spikes (Brader et al., 2007). However, this comes at the cost of the sensitivity of the plasticity rule to the temporal spike patterns as multiple spike patterns might lead to the same binary synaptic weights. In the absence of weight drift as in the proposed hybrid CMOS-memristor architecture, the analog synaptic weights are able to maintain a synaptic trace that better reflects the identity of past spiking patterns (Maass and Markram, 2002).

2.2. Memristive Devices

The memristors that we use as synaptic elements are TiO_{2-x}-based memristors which were fabricated as follows: thermal oxidation was used to grow a 200 nm film of insulating SiO_2 on a 6″ Silicon wafer. Then, bottom electrodes (BEs) were patterned and obtained by conventional optical photolithography, electron beam evaporation and lift-off process. BEs consisted of evaporation of 5 nm adhesive Titanium (Ti) and 10 nm Platinum (Pt) layers. After that, a similar patterning process was used for the 25 nm TiO_{2-x} active layer that was deposited in a Leybold Helios Pro XL Sputterer to achieve high quality film. The film was sputtered from a Titanium metal target with 8 sccm flow of O_2, 35 sccm Ar, 2 kW at the cathode, and 15 sccm O_2, 2 kW at an additional plasma source. Then, again optical photolithography, electron beam evaporation and lift off process were used to pattern and deposit the 10 nm Pt top electrodes (TEs). **Figure 2** shows a cross-section and microphotograph of $Ti/Pt/TiO_{2-x}/Pt$ memristor prototype (device area: $60 \times 60\ \mu$m).

2.3. Circuits for Memristive Learning

The basic building block of the CMOS circuits is a neuron tile which is shown schematically in **Figure 3A**. The tile contains an analog subthreshold leaky integrate and fire neuron which is fully described in Qiao et al. (2015). The neuron integrates synaptic current (with an adjustable leak) on a capacitor. When the capacitor voltage crosses the firing threshold, the neuron generates a digital spike and the capacitor voltage is reset to ground. The plasticity circuit monitors the membrane potential, $Vmem$, and the spike output of the neuron and uses them to evaluate the conditions in Equations (1) and (2). The plasticity circuit internally generates the $C(t)$ signal by low-pass filtering the neuron spikes. The plasticity circuit then generates two digital signals: $'up'$ and $'dn'$ that determine whether incoming synapses/memristors should be potentiated, depressed, or left unchanged when a pre-synaptic spike occurs according to Equations (1) and (2). The plasticity circuit is described in more detail in Qiao et al. (2015).

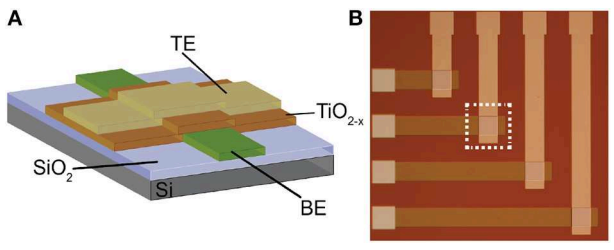


FIGURE 2 | (A) Cross-section and **(B)** microphotograph of the memristive device prototype employed in this work.

A neuron tile has a pre-synaptic and a post-synaptic memristor terminal. These terminals are monitored and driven by the high voltage post- and pre- interfaces which run at a supply voltage of 5 V. All other circuits operate using a 1.8 V supply. The 5 V operation allows the memristor interface circuits to apply higher voltage pulses to the memristor terminals. The memristor conductance changes if pulses above a certain magnitude (the write threshold) are applied across it. The direction of the change depends on the polarity of the pulse. We designed the interface circuits so that they can interface to memristors having resistance values as low as 1 KOhm and deliver write pulses of either polarity with an amplitude of up to 2 V. The write voltage threshold for the memristor devices we use in this paper is much lower than 2 V. The height of the write pulses are programmable, however, so we can control their amplitudes up to 2 V. The width of the programming pulse is also configurable and can be as wide as 1 ms. The read pulse amplitude (which needs to be below the write threshold) is adjustable in the 0–2 V range and its width is also adjustable. The memristor is inserted between the pre-synaptic terminal of one tile and the post-synaptic terminal of another (or the same) tile. Spikes generated in the neuron circuit of the pre-synaptic tile will then cause a current proportional to the memristor conductance to be injected into the post-synaptic tile neuron. Moreover, based on the output of the plasticity circuit in the post-synaptic tile, a voltage pulse of the appropriate polarity is applied across the memristor terminals to increase/potentiate or decrease/depress its conductance when the pre-synaptic neuron tile generates a spike. In the rest of this section, we describe how this behavior is realized.

The pre- and post-synaptic memristor interfaces are shown in more detail in **Figure 3B** where they are linked by a memristive element. We retain the ability to disconnect the post-synaptic circuit from the memristor post-synaptic terminal using switch S1. The pre-synaptic memristor terminal is kept floating by default so no current can flow through the memristor and its value remains constant. The post-synaptic terminal is monitoring the current flowing through the memristor and injecting a proportional current into the neuron. By keeping the pre-synaptic terminal floating, no current flows through the memristor, and no current is injected into the post-synaptic neuron. When the pre-synaptic tile neuron spikes, or when the tile receives an AER event from off-chip, the pre-synaptic terminal is strongly clamped at 2.5 V for a short duration that is controlled by an analog bias. By clamping the pre-synaptic terminal to the middle of the supply voltage, we are able to apply pulses of either polarity with an amplitude of up to 2.5 V by setting the appropriate voltage on the post-synaptic terminal. If the post-synaptic terminal is clamped to V_{post}, then on pre-synaptic spike, a pulse of amplitude $V_{post} - 2.5$ is applied across the memristor. Assume switch S1 is closed. The post-synaptic terminal can be clamped to one of three possible values: 4.5, 0.5, or 3 V. These clamping voltages can be adjusted through analog biases. The clamping is done by the strong transistors M1 and

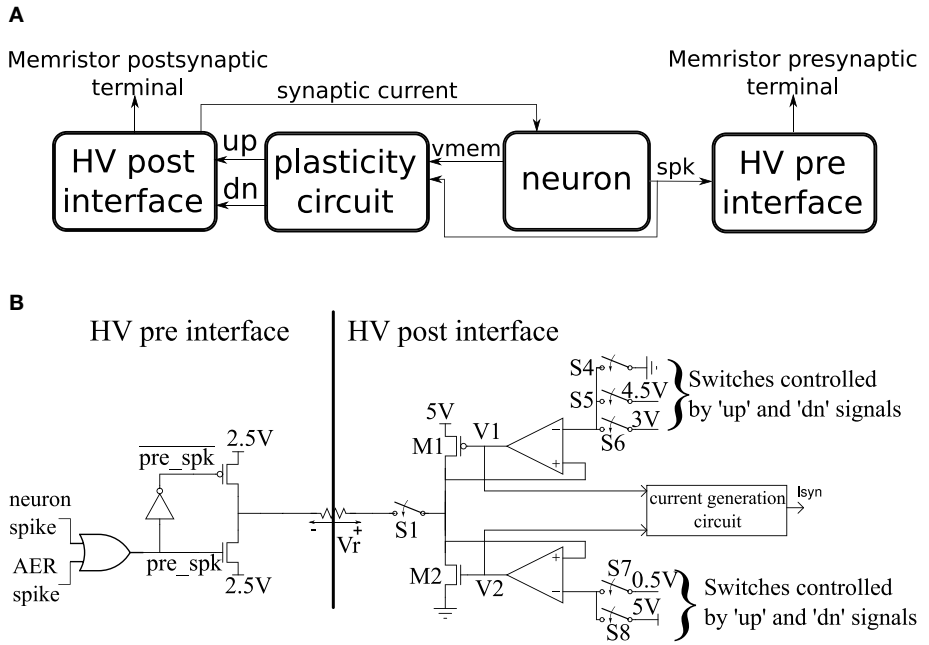


FIGURE 3 | (A) High-level schematic of a neuron tile. **(B)** More detailed schematics of the high voltage pre- and post-synaptic interfaces in the neuron tile.

M2 which are each part of a negative feedback loop that controls their gate potentials so as to maintain their drain potentials at one of the three voltage clamp values. A number of switches which are controlled by the $'up'$ and $'dn'$ signals from the plasticity circuit determine which clamping voltage is selected according to **Table 1**. For example, if switches S5 and S8 are closed and switches S4, S6, and S7 are open, the post-synaptic terminal is clamped by a PFET at 4.5 V. Switches S4–S8 are implemented as single transistors as each switch has to pass a bias voltages that is always either above 2.5 (PFET is used) or below 2.5 (NFET is used). Switch S1 is implemented as a transmission gate.

At a pre-synaptic spike which causes the pre-synaptic terminal to be clamped to 2.5 V, the memristor experiences a voltage pulse of either 2.0, −2.0, or 0.5 V depending on whether the post-synaptic terminal is at 4.5, 0.5, or 3V respectively. These three cases can either potentiate/increase the memristor conductance, depress/decrease it, or leave it unchanged respectively. It is the plasticity circuit, which through the $'up'$ and $'dn'$ signals controls switches S4–S8, which chooses between these three cases (**Table 1**).

The post-synaptic side indirectly senses the memristor conductance from the gate voltages V1 and V2. When the pre-synaptic side is floating, the two feedback loops push V1 and V2 to 5V and 0V, respectively. The current generation circuit will then generate very little current. At a pre-synaptic event, either V1 or V2 abruptly changes so that the actively clamping transistor has increased effective gate-source voltage so as to be able to source/sink the memristor current while maintaining the drain terminal at the clamp voltage. Larger memristor conductance translates to a larger change in V1 or V2 and based on this change, a proportional current *Isyn* is generated and injected into the post-synaptic neuron. The current generation circuit

TABLE 1 | Effect of $'up'$ and $'dn'$ signals on the post-synaptic terminal potential which in turn determines the type of plasticity event induced on pre-synaptic spikes.

Plasticity signal	Vpost	Plasticity event	Open switches	Closed switches
$'up'$ = 0 and $'dn'$ = 0	3.0	No change	S4, S5, S7	S6, S8
$'up'$ = 1 and $'dn'$ = 0	4.5	Potentiate	S4, S6, S7	S5, S8
$'up'$ = 0 and $'dn'$ = 1	0.5	Depress	S8, S5, S6	S4, S7

Shown are the open and closed switches in each case. The switches are controlled by the $'up'$ and $'dn'$ signals.

approximately implements the equation:

$$Isyn = A * V2 - B * V1 \tag{6}$$

Where A and B are constants adjusted through biases. This linear equation is, however, valid in a limited regime of V_1 and V_2. This regime can be adjusted through biases. Note that *Isyn* is proportional to the absolute value of the memristor current, regardless of whether the current is sourced by transistor M1 or sunk by transistor M2. The $'up'$ and $'dn'$ signals can not both be high at the same time. For the three possible configurations of the $'up'$ and $'dn'$ signals in **Table 1**, M1 and M2 can not be supplying current at the same time. For the possible configuration of switches shown in **Table 1**, the feedback loops controlling V1 and V2 ensure that the gate-source voltage of M1 and M2 can not be simultaneously non-zero. This guarantees that the current in either M1 or M2 is the current flowing through the memristor.

This active clamp technique allows maximum voltage headroom for transistors M1 and M2 which allows them to clamp the post-synaptic terminal at voltages near the supply rails. It

also enables precise control over the magnitude of the voltage pulses applied across the memristor. In Section 3.2, we present experimental results to illustrate the behavior of the circuit in **Figure 3A**.

The synaptic weight in the original spike-based Perceptron learning rule has only two stable states due to the weight drift (Equations 4 and 5) which pushes the weight to either a high or a low value. This mechanism is not present in our architecture; the synaptic weight (memristor conductance) is an analog quantity that only changes in response to pre-synaptic spikes and is stable otherwise. Realizing analog synaptic weights that are long-term stable is difficult in pure CMOS as analog weights that are encoded using charge on a capacitor are easily corruptible through leakage paths and capacitive coupling to nearby nodes. Therefore, in pure CMOS, a multi-stability mechanism is required to push the weights to well-defined and stable discrete states. Hybrid CMOS-memristor architectures like ours can realize naturally stable analog weights (memristor conductances) and thus do not require such a mechanism.

An 8 * 8 array of the neuron tile shown in **Figure 3** was fabricated on a standard 6 M 180 nm CMOS process as part of a larger multi-purpose neuromorphic chip shown in **Figure 4A**. The chip contains a bias generator based on the design in Delbruck and Lichtsteiner (2006). The bias generator has a low pin-count (5 pins) digital programming interface that can be used to set the values of the analog biases used in the neuron tile array. The other components of the multi-purpose neuromorphic chip are described in detail in Qiao et al. (2015) but they are not relevant for the current paper. Address event representation (AER) interfaces carry spikes to/from the neuron tile array. The pre- and post-synaptic terminals of the 64 neuron tiles were routed to the top-metal level to make it possible to directly deposit a cross-bar array of TiO_{2-x} memristors on top that connects a memristor between each pre-synaptic terminal and each post-synaptic terminal. This post-processing step was not carried out. In the chip, the pre- and post-synaptic memristor terminals of two neuron tiles were directly connected to pads. An off-chip memristor was then connected between the pre-synaptic terminal of one of these neuron tiles and the post-synaptic terminal of the other tile as shown in **Figure 4B**. This setup was used to obtain the measurements presented in the rest of this paper.

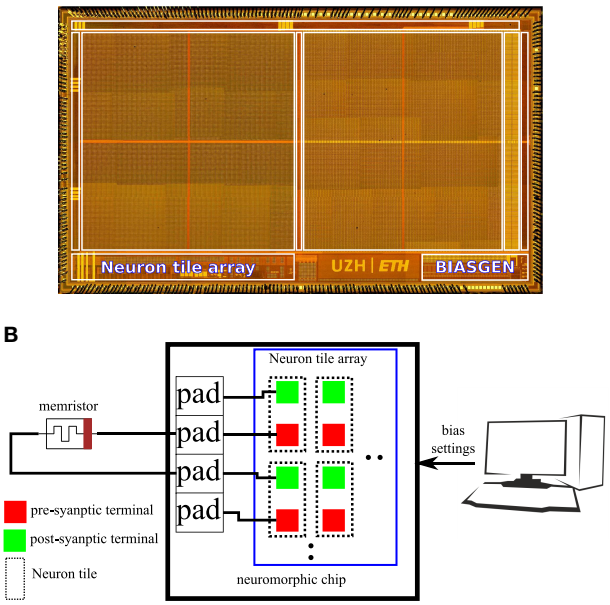


FIGURE 4 | (A) Micrograph of the multi-purpose neuromorphic chip die showing the neuron tile array and the bias generator. **(B)** Illustration of the hardware setup used to obtain measurements for this paper. The pre-synaptic terminal of one neuron tile is connected to the post-synaptic terminal of another neuron tile through an off-chip TiO_{2-x} memristor. A PC controls the digital settings of the on-chip bias generator.

2.4. Crossbar Operation

In a crossbar configuration, N neuron tiles are interconnected by an array of N^2 memristors where there is a memristor connecting each pre-synaptic terminal to each post-synaptic terminal. To achieve high synaptic/memristor integration densities, it is important to avoid memristor specific CMOS circuits and only access the memristors through the N pre-synaptic terminals and N postsynaptic terminals which form the row lines and column lines of the $N * N$ memristor array. Consider the simple case of $N = 2$ neuron tiles connected using $N^2 = 4$ memristors. If the post-interfaces in tile 1 and 2 are clamping the post-synaptic terminals to different voltages (which would be the case if one of them is in the $'up'$ state and the other is in the $'dn'$ state) then current would flow between the post-synaptic terminals through two memristors connected in series. This would lead to changes in the conductances of these memristors in the absence of pre-synaptic spikes and to synaptic current being mistakenly injected into the neurons. Crossbar operation is thus only possible if plasticity is switched off through the analog biases so that all post-synaptic terminals are clamped at the same potential.

One benefit of using a crossbar array to implement synaptic matrices is that a post-synaptic neuron only needs to know the aggregate input it receives from all synapses rather than the individual contributions. This considerably relaxes the design of the driver circuits as these circuits do not need to isolate the contribution of single devices. Moreover, it has been shown that small selector-less arrays can perform quite well even as analog memory, where good isolation of the contribution of each individual element is required, provided certain assumptions about the switching characteristics of the memristors (e.g., concerning maximum and minimum resistive states) hold (Serb et al., 2015). Thus, even though the current implementation does not take the additional complications of crossbar configurations into account, there is evidence that extension of our work to first small, selector-less arrays and then potentially to larger selector-based arrays is possible.

3. Results

3.1. Initial Memristor Programming Results

Before the memristors could be used as artificial synapses they were electrically prepared for operation. The preparation

procedure consisted of an electroforming step, a stabilization period and a characterization stage. During all stages the devices are biased by a series of square-wave pulses of fixed duration (100 μ s) and variable amplitude.

Initially, the measured resistive state (RS) of all our devices was above the 10 MΩ mark. During electroforming devices were subjected to voltage pulse ramps beginning at 1 V and increasing in steps of 0.5 V until the RS dropped to below 500 kΩ or the maximum limit of 8 V was reached. Typically, electroforming was achieved after applying a 6 V pulse. During electroforming, voltage was applied to our devices through a 100 kΩ series resistor as a measure to protect them against unduly high power dissipation and consequent damage.

In the stabilization period devices are subjected to pulse trains whose amplitudes and polarities are determined by trial and error. During this phase the devices are forced to oscillate between more resistive and more conductive states. This is achieved through use of a bipolar stimulation protocol, that is pulses of opposite polarities drive the RS in opposite directions. During the stabilization period no stable limits for the operational RS ceiling and floor can be reliably determined, nor can appropriate voltages be found at which the memristor will reliably switch between floor and ceiling.

The characterization stage follows seamlessly from the stabilization period as the device settles to an operational RS range. In this phase voltage pulse amplitudes are trimmed until a set of amplitudes for normal operation biasing is selected. **Figure 5** shows a typical characterization stage series of read-outs obtained from a well-behaved device. Notably different voltage values are tested for both SET-type (toward lower RS) and RESET-type pulse polarities within a relatively narrow range (≈ 200 mV). Typically devices can operate comfortably within such narrow ranges although their operational range and the number of pulses it takes to transition between floor and ceiling (and vice versa) will be affected by the exact choice of pulse voltage. See **Figure 5** for an example.

Once pulsing voltages have been determined, the memristor may be connected to the appropriately configured neuromorphic circuitry, ready for bipolar-mode operation.

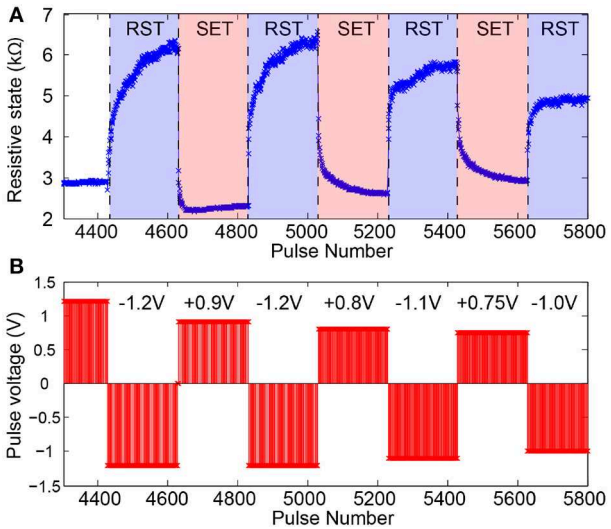


FIGURE 5 | Memristor switching characteristics during SET-type and RESET-type (RST) pulsing. (A) Evolution of device under test (DUT) resistive state in reaction to pulsed input stimulation. **(B)** Corresponding pulsing sequence. Pulse width fixed at 100 μs.

3.2. Characterization of CMOS Plasticity Circuits

On each pre-synaptic spike, the pre- and post-synaptic interface circuits in the neuron tile shown in **Figure 3** apply a voltage pulse to update the memristor value according to the spike-based perceptron learning rule. This behavior is illustrated in **Figure 6A** where a fixed resistor was inserted between the pre-synaptic terminal of one tile and the post-synaptic terminal of another (as in **Figure 4B** but using a resistor instead of a memristor). Constant current is injected into the neurons to maintain a constant firing rate. The calcium signal, $C(t)$, jumps up after each spike and enters the plasticity range, then it decays back out of the plasticity range. The bottom two plots in **Figure 6A** show the $'up'$ and $'dn'$ signals and the neuron membrane potential in the post-synaptic tile. The $'up'$ and $'dn'$ signals are generated by the plasticity circuit in the post-synaptic tile (see **Figure 3A**). This plasticity circuit calculates the calcium variable, $C(t)$, from the post-synaptic neuron spikes according to Equation (3). It evaluates the conditions in Equations (1) and (2) to decide whether to potentiate, depress, or leave unchanged incoming synapses when the pre-synaptic neuron spikes. This decision is communicated to the post-synaptic interface circuit which clamps the post-synaptic terminal voltage $vpost$ at $4.1V$ (when the $'up'$ signal is high), 0.1 V (when the $'dn'$ signal is high), or 3 V (when both the $'dn'$ and $'up'$ are low) as shown in **Figure 6A**. The pre-synaptic terminal is floating by default and is clamped at 2.5 V for a short duration on each pre-synaptic spike. For each pre-synaptic spike, this causes $vpost - vpre$ to be approximately 2 V when the $'up'$ signal is high which would increase the memristor conductance (potentiation), –2 V when the $'dn'$ signal is high which would decrease the memristor conductance (depression), and 0.5 V otherwise as shown in **Figure 6A** which would leave the memristor conductance unchanged and simply read out its value. In **Figure 6A**, at the first pre-synaptic spike, $C(t)$ is outside the plasticity range and a small read pulse is applied. The subsequent pre-synaptic spikes arrive first in the depression, then in the potentiation intervals of the post-synaptic tile and large amplitude pulses with the appropriate polarity are applied.

Figure 6B shows how the firing frequency of the post-synaptic neuron varies as a function of the value of the resistor connecting it to the pre-synaptic tile. The pre-synaptic tile generates spikes at a constant frequency. The firing frequency of the post-synaptic neuron steadily decreases with the decreasing conductance of the resistor. The bias conditions were chosen to obtain a linear region in the 1–4 KOhm resistance range beyond which the post-synaptic neuron firing frequency saturates at a lower bound. The neuron is biased to have a spontaneous baseline firing rate which is about 290 Hz. The transfer function from the synaptic resistance to the synaptic current injected into the neuron is linear in the 1–4 KOhm region in **Figure 6B** but beyond that, it is highly non-linear causing a small increase in synaptic resistance

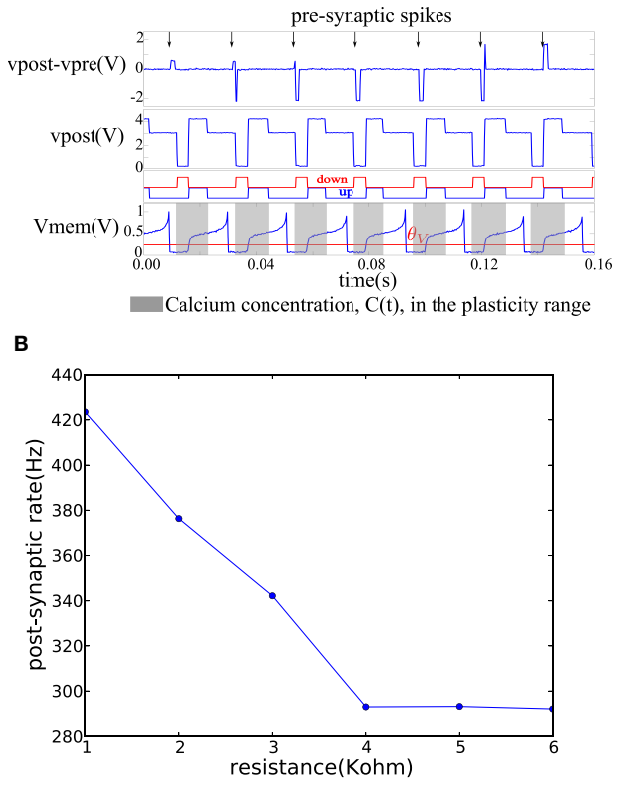


FIGURE 6 | (A) Membrane potential and the *'up'* and *'dn'* signals of a post-synaptic tile receiving spikes at a constant rate from a pre-synaptic tile. The plasticity circuit controls the post-synaptic terminal potential so as to generate an appropriate programming/read pulse across the *vpost* − *vpre* terminals on each pre-synaptic spike. **(B)** Decrease of post-synaptic neuron firing frequency as the conductance of the afferent synaptic element decreases while the pre-synaptic neuron firing rate is kept constant.

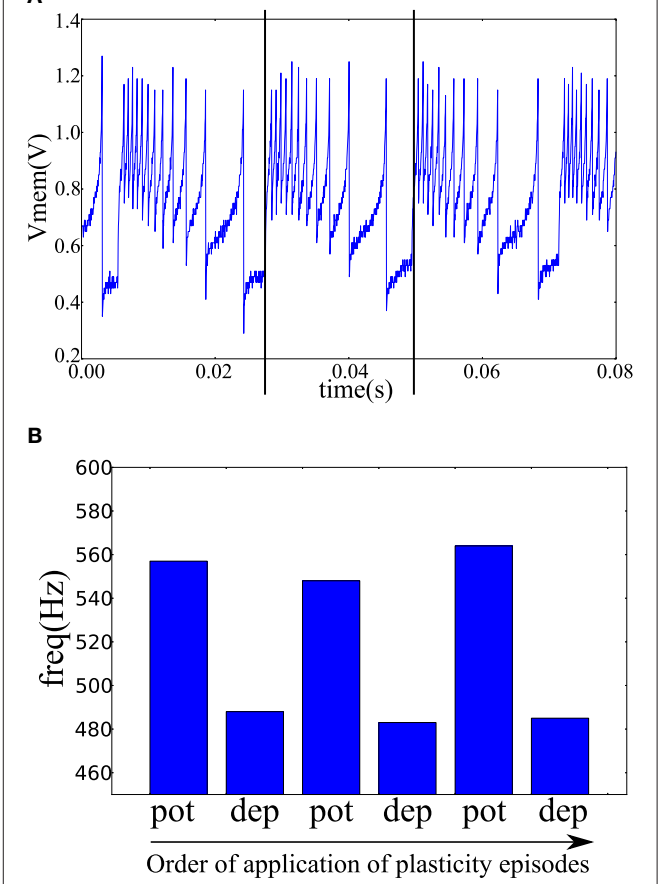


FIGURE 7 | (A) The post-synaptic membrane potential in response to a pre-synaptic pulse train of 47 Hz. Vertical lines indicate the times of two pre-synaptic spikes. **(B)** Frequency of the post-synaptic neuron after a number of plasticity episodes. Plasticity episodes were applied in succession by adjusting the analog biases to put the post-synaptic neuron tile in the potentiation and depression regimes.

to lead to a greatly reduced synaptic current which becomes negligible compared to the constant injection current used to maintain the baseline firing rate. The neuron then saturates at the baseline firing rate.

3.3. Memristive Plasticity Experiments

In the fabricated chip, the pre- and post-synaptic terminals of two neuron tiles were available on the chip pads. The pre-synaptic terminal of one tile was connected to the post-synaptic terminal of the other tile through a TiO_{2-x} memristor as shown in **Figure 4B**. The pre-synaptic neuron was biased to fire at 47 Hz. The spike-based perceptron learning circuit was then successively cycled between the potentiation and depression regimes (using the analog biases) and the resulting post-synaptic firing rate was observed. The post-synaptic firing rate was taken as an indication of the synaptic weight or the conductance of the memristive element. The system showed correct operation with the postsynaptic firing rate increasing after a potentiation episode and decreasing after a depression episode as shown in **Figure 7**.

A potentiation episode involves setting the biases of the plasticity circuit in **Figure 3A** so that its *'up'* output signal is constantly high which will cause the memristor post-synaptic terminal to be clamped at approximately 4.1 V as shown in **Figure 6A**. On each pre-synaptic spike, a pulse of approximately 2 V is thus applied across the memristor terminals which will act to increase its conductance. Similarly, in a depression episode, the *'dn'* output signal is constantly high which will cause a pulse of approximately −2 V to be applied across the memristor terminals on each pre-synaptic spike which will act to decrease its conductance. After each plasticity episode, plasticity was disabled and the post-synaptic neuron firing rate measured, then the next plasticity episode is applied.

4. Discussion

4.1. Memristive Device Characteristics

Figure 5 shows the typical operation of a "well-behaved" memristor in response to trains of input voltage pulses. A number of key features are noteworthy:

- *Bipolar operation:* Pulses of opposite polarity precipitate resistive state changes in opposite directions. In the case of our

devices, a positive voltage applied to the top electrode (bottom electrode grounded) causes potentiation.

- *Bidirectionally gradual switching:* Transitions between resistive state floor and ceiling occur over many pulses, not just one. This allows the device to work as a multi-level weight artificial synapse (as opposed to binary).
- *Bidirectionally saturating switching:* When a device is bombarded by trains of identical voltage pulses it approaches its operational resistive state floor and ceiling in progressively smaller steps. This implies that the middle of the resistive state range is expected to be most often unoccupied *in operando*, as it is traversed quickly in either direction under pulsing. The resistive state will be therefore multi-level in nature, but most of the time distinctly high or low.
- *Biasing parameter variation tolerance:* The device can remain functional under a relatively wide range of bias voltages. We obtain good switching behavior for voltage pulses in the 0.75–1.2 V range. The device can safely operate with voltage pulses of up to 2 V. This bodes well for operation in tandem with practical electronic systems and for resistive switching behavior tuning.

These features allow the memristive devices to exhibit the correct behavior when coupled to the neuromorphic circuits described in Section 2.3, both as binary and as multi-level synapses. Only binary synaptic operation was investigated in the plasticity experiments.

4.2. The Spike-based Perceptron Learning Rule in CMOS-memristor Architectures

The spike-based Perceptron plasticity rule has been implemented in CMOS neuromorphic systems using various types of circuits such as subthreshold circuits (Mitra et al., 2009) and switched capacitor circuits (Noack et al., 2015). In this paper, we have presented a physical implementation of the first hybrid CMOS-memristor architecture that implements a spike-based Perceptron learning plasticity rule. The physical CMOS-memristor system we presented is a standalone system in which the custom CMOS chip connects directly to the memristive devices. The CMOS chip implements the neuron elements together with dedicated per-neuron circuits that can program (potentiate or depress) the memristive synaptic elements as well as sense their conductances/weights to generate proportional Excitatory Post-Synaptic Currents (EPSCs) in the post-synaptic neuron in response to pre-synaptic spikes. We have presented direct measurements that illustrate the behavior of this physical CMOS-memristor system. This is the first standalone neuromorphic system that combines custom neuron circuits with memristor programming and sensing circuits acting on physical memristive devices.

Many highly accurate and biologically grounded, i.e., non-empirical, synaptic plasticity rules make use of several auxiliary variables beyond spike times in the pre- and post-synaptic neurons to control synaptic weight updates (Pfister et al., 2006; Brader et al., 2007; Clopath and Gerstner, 2010; Mayr and Partzsch, 2010; Graupner and Brunel, 2012). These auxiliary variables may include low-pass filtered versions of the membrane potential (Clopath and Gerstner, 2010) or a low-pass filtered version of the neuron's spike train (Brader et al., 2007). Interestingly, the time difference between pre- and post-synaptic spikes does not figure explicitly in these models. This presents a problem for current neuromorphic memristive architectures that mainly depend on this time difference (through the overlap between pre- and post-synaptic spike-triggered waveforms) to induce weight updates. These architectures will not be able to handle weight updates that are triggered on single pre- or post-synaptic spikes.

The architecture we presented triggers weight updates on single pre-synaptic spikes. This has a significant advantage: at the time of a pre-synaptic spike, the neuromorphic synapse can be immediately potentiated or depressed based on the current state of the post-synaptic neuron; the neuromorphic system does not have to wait for a post-synaptic spike to know the outcome of the plasticity event. Implementations of classical pair-wise STDP rules using memristors typically trigger long waveforms on the pre- and post-synaptic sides of the memristor in response to pre- and post-synaptic spikes respectively. When these waveforms overlap, the potential difference across the memristor exceeds a threshold and changes in memristor conductance occur. The duration of these waveforms dictate the STDP window. The overlapping waveforms paradigm is problematic in the high spike rate regime as multiple spikes can occur within the STDP window, thereby corrupting the synaptic weight update. By contrast this problem is completely avoided in the case of the spike-based Perceptron learning rule.

In the original learning rule (Brader et al., 2007) the weights were bistable, i.e., they gradually drifted to one of two stable states. This had the effect of consolidating synaptic changes and making it more difficult for a synaptic pattern to be corrupted by spurious spikes. Our architecture does not implement such continuous (non event-driven) weight drift. This indicates that synaptic rule features that simplify pure CMOS implementations like bistable weights do not necessarily translate to simpler CMOS-memristor implementations.

4.3. Outlook

The architecture we describe represents a first step toward hybrid CMOS-memristor implementations of more elaborate plasticity rules that go beyond standard STDP. Further developments will have to address the problem of plastic crossbar operation as well as mechanisms that allow continuous or non event-driven weight updates.

Acknowledgment

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

Abbott, L. F., and Regehr, W. G. (2004). Synaptic computation. *Nature* 431, 796–803. doi: 10.1038/nature03010

Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran A. R., Bussat, J., et al. (2014). Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* 102, 699–716. doi: 10.1109/jproc.2014.2313565

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

Binzegger, T., Douglas, R. J., and Martin, K. (2004). A quantitative map of the circuit of cat primary visual cortex. *J. Neurosci.* 24, 8441–8453. doi: 10.1523/JNEUROSCI.1400-04.2004

Brader, J. M., Senn, W., and Fusi, S. (2007). Learning real world stimuli in a neural network with spike-driven synaptic dynamics. *Neural Comput.* 19, 2881–2912. doi: 10.1162/neco.2007.19.11.2881

Chicca, E., Stefanini, F., Bartolozzi, C., and Indiveri, G. (2014). Neuromorphic electronic circuits for building autonomous cognitive systems. *Proc. IEEE* 102, 1367–1388. doi: 10.1109/JPROC.2014.2313954

Chua, L. (1971). Memristor-the missing circuit element. *Circ. Theory IEEE Trans.* 18, 507–519. doi: 10.1109/TCT.1971.1083337

Clopath, C., and Gerstner, W. (2010). Voltage and spike timing interact in stdp – a unified model. *Front. Synaptic Neurosci.* 2:25. doi: 10.3389/fnsyn.2010.00025

Delbruck, T., and Lichtsteiner, P. (2006). "Fully programmable bias current generator with 24 bit resolution per bias," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings 2006. IEEE International Symposium* (island of Kos: IEEE), 4.

Du, N., Manjunath, N., Shuai, Y., Bürger, D., Skorupa, I., Schüffny, R., et al. (2014). Novel implementation of memristive systems for data encryption and obfuscation. *J. Appl. Phys.* 115, 124501. doi: 10.1063/1.4869262

FACETS. (2005–2009). *Fast Analog omputing with Emergent Transient States in Neural Architectures (FACETS).* FP6-2005-015879 EU Grant. Heidelberg.

Fusi, S., Annunziato, M., Badoni, D., Salamon, A., and Amit, D. (2000). Spike–driven synaptic plasticity: theory, simulation, VLSI implementation. *Neural Comput.* 12, 2227–2258. doi: 10.1162/089976600300014917

Graupner, M., and Brunel, N. (2012). Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location. *Proc. Natl. Acad. Sci. U.S.A.* 109, 3991–3996. doi: 10.1073/pnas.1109359109

Indiveri, G., Stefanini, F., and Chicca, E. (2010). "Spike-based learning with a generalized integrate and fire silicon neuron," in *International Symposium on Circuits and Systems, (ISCAS), 2010* (Paris: IEEE), 1951–1954. Available online at: http://ncs.ethz.ch/pubs/pdf/Indiveri_etal10.pdf

Indiveri, G., Linares-Barranco, B., Hamilton, T. J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5:73. doi: 10.3389/fnins.2011.00073

Indiveri, G., Linares-Barranco, B., Legenstein, R., Deligeorgis, G., and Prodromakis, T. (2013). Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* 24:384010. doi: 10.1088/0957-4484/24/38/384010

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301. doi: 10.1021/nl904092h

Lin, Z., and Wang, H. (2010). Efficient image encryption using a chaos-based pwl memristor. *IETE Tech. Rev.* 27, 318–325. doi: 10.4103/0256-4602.64605

Linn, E., Rosezin, R., Tappertzhofen, S., Böttger, U., and Waser, R. (2012). Beyond von neumann - logic operations in passive crossbar arrays alongside memory perations. *Nanotechnology* 23:305205. doi: 10.1088/0957-4484/23/30/305205

Lisman, J., and Spruston, N. (2010). Questions about stdp as a general model of synaptic plasticity. *Front. Synaptic Neurosci.* 2:140. doi: 10.3389/fnsyn.2010.00140

Maass, W., and Markram, H. (2002). Synapses as dynamic memory buffers. *Neural Netw.* 15, 155–161. doi: 10.1016/S0893-6080(01)00144-7

Mayr, C. G., and Partzsch, J. (2010). Rate and pulse based plasticity governed by local synaptic state variables. *Front. Synaptic Neurosci.* 2:28. doi: 10.3389/fnsyn.2010.00033

Mayr, C., Stärke, P., Partzsch, J., Cederstroem, L., Schüffny, R., Shuai, Y., et al. (2012). "Waveform driven plasticity in BiFeO3 memristive devices: model and implementation," in *Advances in Neural Information Processing Systems 25* (Lake Tahoe, CA), 1700–1708.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642

Mitra, S., Fusi, S., and Indiveri, G. (2009). Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI. *Biomed. Circ. Syst. IEEE Trans.* 3, 32–42. doi: 10.1109/TBCAS.2008.2005781

Moreno, C., Munuera, C., Valencia, S., Kronast, F., Obradors, X., and Ocal, C. (2010). Reversible resistive switching and multilevel recording in La0.7Sr0.3MnO3 thin films for low cost nonvolatile memories. *Nano Lett.* 10, 3828–3835. doi: 10.1021/nl1008162

Navaridas, J., Furber, S., Garside, J., Jin, X., Khan, M., Lester, D., et al. (2013). Spinnaker: fault tolerance in a power-and area-constrained large-scale neuromimetic architecture. *Parallel Comput.* 39, 693–708. doi: 10.1016/j.parco.2013.09.001

Ning, Q., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., et al. (2015). A re-configurable on-line learning spiking neuromorphic processor. *Front. Neurosci.* 9:141. doi: 10.3389/fnins.2015.00141

Noack, M., Partzsch, J., Mayr, C. G., Hänzsche, S., Scholze, S., Höppner, S., et al. (2015). Switched-capacitor realization of presynaptic short-term-plasticity and stop-learning synapses in 28 nm CMOS. *Front. Neurosci.* 9:10. doi: 10.3389/fnins.2015.00010

Pfister J. P., Toyoizumi, T., Barber, D., and Gerstner, W. (2006). Optimal spike-timing dependent plasticity for precise action potential firing in supervised learning. *Neural Comput.* 18, 1309–1339. doi: 10.1162/neco.2006.18.6.1318

Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., et al. (2015). A re-configurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Front. Neurosci.* 9:141. doi: 10.3389/fnins.2015.00141

Saighi, S., Mayr, C., Linares-Barranco, B., Serrano-Gotarredona, T., Schmidt, H., Lecerf, G., et al. (2015). Plasticity in memristive devices. *Front. Neurosci.* 9:51. doi: 10.3389/fnins.2015.00051

Schemmel, J., Brüderle, D., Meier, K., and Ostendorf, B. (2007). "Modeling synaptic plasticity within networks of highly accelerated I&F neurons," in *International Symposium on Circuits and Systems, (ISCAS), 2007* (New Orleans, LA: IEEE), 3367–3370.

Schemmel, J., Grübl, A., Hartmann, S., Kononov, A., Mayr, C., Meier, K., et al. (2012). "Live demonstration: a scaled-down version of the BrainScaleS wafer-scale neuromorphic system," in *IEEE International Symposium on Circuits and Systems ISCAS 2012* (Seoul), 702.

Senn, W., and Fusi, S. (2005). Learning only when necessary: better memories of correlated patterns in networks with bounded synapses. *Neural Comput.* 17, 2106–2138. doi: 10.1162/0899766054615644

Serb, A., Redman-White, W., Papavassiliou, C., Berdan, R., and Prodromakis, T. (2015). "Limitations and precision requirements for read-out of passive, linear, selectorless rram arrays," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium* (Lisbon: IEEE), 189–192.

Serrano-Gotarredona, T., Masquelier, T., Prodromakis, T., Indiveri, G., and Linares-Barranco, B. (2013). STDP and STDP variations with memristors for spiking neuromorphic learning systems. *Front. Neurosci.* 7:2. doi: 10.3389/fnins.2013.00002

Shuai, Y., Ou, X., Luo, W., Du, N., Wu, C., Zhang, W., et al. (2013). Nonvolatile multilevel resistive switching in Ar+ irradiated BiFeO3 thin films. *IEEE Electron Device Lett.* 34, 54–56. doi: 10.1109/LED.2012.2227666

Sjöström, P., Turrigiano, G., and Nelson, S. (2001). Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* 32, 1149–1164. doi: 10.1016/S0896-6273(01)00542-6

Sjöström, P., Rancz, E., Roth, A., and Häusser, M. (2008). Dendritic excitability and synaptic plasticity. *Physiol. Rev.* 88, 769–840. doi: 10.1152/physrev.00016.2007

Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. (2008). The missing memristor found. *Nature* 453, 80–83. doi: 10.1038/nature06932

You, T., Shuai, Y., Luo, W., Du, N., Bürger, D., Skorupa, I., et al. (2014). Exploiting memristive BiFeO3 bilayer structures for compact sequential logics. *Adv. Funct. Mater.* 24, 3357–3365. doi: 10.1002/adfm.201303365

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 30 June 2015
doi: 10.3389/fnins.2015.00227

Single pairing spike-timing dependent plasticity in $BiFeO_3$ memristors with a time window of 25 ms to 125 μs

Nan Du[1], Mahdi Kiani[1], Christian G. Mayr[2]*, Tiangui You[1], Danilo Bürger[1], Ilona Skorupa[1,3], Oliver G. Schmidt[1,4] and Heidemarie Schmidt[1]**

[1] Material Systems for Nanoelectronics, Faculty of Electrical and Information Engineering, Chemnitz University of Technology, Chemnitz, Germany, [2] Neuromorphic Cognitive Systems Group, Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland, [3] Semiconductor Materials, Institute of Ion Beam Physics and Materials Research, HZDR Innovation GmbH, Dresden, Germany, [4] Institute for Integrative Nanosciences, IFW Dresden, Dresden, Germany

Memristive devices are popular among neuromorphic engineers for their ability to emulate forms of spike-driven synaptic plasticity by applying specific voltage and current waveforms at their two terminals. In this paper, we investigate spike-timing dependent plasticity (STDP) with a single pairing of one presynaptic voltage spike and one post-synaptic voltage spike in a $BiFeO_3$ memristive device. In most memristive materials the learning window is primarily a function of the material characteristics and not of the applied waveform. In contrast, we show that the analog resistive switching of the developed artificial synapses allows to adjust the learning time constant of the STDP function from 25 ms to 125 μs via the duration of applied voltage spikes. Also, as the induced weight change may degrade, we investigate the remanence of the resistance change for several hours after analog resistive switching, thus emulating the processes expected in biological synapses. As the power consumption is a major constraint in neuromorphic circuits, we show methods to reduce the consumed energy per setting pulse to only 4.5 pJ in the developed artificial synapses.

Keywords: $BiFeO_3$ memristor, artificial synapse, single pairing STDP, memory consolidation, learning window, low-power device

OPEN ACCESS

Edited by:
Themis Prodromakis, University of Southampton, UK

Reviewed by:
Siddharth Joshi, University of California, San Diego, USA
Joaquin Sitte, Queensland University of Technology, Australia

****Correspondence:***
Nan Du, Material Systems for Nanoelectronics, Faculty of Electrical and Information Engineering, Chemnitz University of Technology, Reichenhainer Str. 39/41, 09126 Chemnitz, Germany nan.du@s2012.tu-chemnitz.de; Christian G. Mayr, Neuromorphic Cognitive Systems Group, Institute of Neuroinformatics, University of Zurich and ETH Zurich, Winterthurerstr. 190, CH-8057 Zurich, Switzerland christian.mayr@tu-dresden.de; Heidemarie Schmidt, Material Systems for Nanoelectronics, Faculty of Electrical and Information Engineering, Chemnitz University of Technology, Reichenhainer Str. 39/41, 09126 Chemnitz, Germany heidemarie.schmidt@etit.tu-chemnitz.de

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *19 February 2015*
Accepted: *11 June 2015*
Published: *30 June 2015*

Citation:
Du N, Kiani M, Mayr CG, You T, Bürger D, Skorupa I, Schmidt OG and Schmidt H (2015) Single pairing spike-timing dependent plasticity in $BiFeO_3$ memristors with a time window of 25 ms to 125 μs Front. Neurosci. 9:227. doi: 10.3389/fnins.2015.00227

Introduction

Since the discovery of spike-timing dependent plasticity (STDP) in biological synapses (Bi and Poo, 1998; Snider, 2008; Di Lorenzo and Victor, 2013), scientists have been captivated by the idea of changing the synaptic weight, i.e., the strength between the pre- and post-neuron, in bioinspired electronic systems in a fashion similar to biology (Indiveri et al., 2006). However, the circuit-oriented approach is complicated because the "synaptic weight" variable has to be stored typically either as charge in a capacitor (Koickal et al., 2006) or even digitally in neuromorphic IC (Schemmel et al., 2012; Mayr et al., 2013). This adds circuit complexity and increases energy consumption (Indiveri et al., 2006; Adee, 2009; Ananthanarayanan et al., 2009). Therefore, nonvolatile analog resistive switches, namely resistive random-access memory (RRAM) or memristors (Chua, 1971; Du et al., 2013), responding to well-defined input signals by suitably changing their internal state ("weight") are currently developed. For example, the emulation of STDP with 60–80 pairings of

pre- and post-synaptic spikes has been shown for artificial synapses based on memristive TiO_x (Seo et al., 2011; Thomas and Kaltschmidt, 2014), WO_x (Chang et al., 2011), HfO_x (Yu et al., 2011), GST (Kuzum et al., 2012), and on the memristive $BiFeO_3$ (Mayr et al., 2012; Cederström et al., 2013).

Figure 1A shows a memristor between the electrical Integrate & Fire (I&F) neurons. The synaptic weight of the memristor can be controlled by the time delay Δt between pre- and post-spike from the 1st layer I&F neuron (**Figure 1A**) (Zamarreño-Ramos et al., 2011). The 2nd layer I&F neuron sums up the signals from all incoming neurons and generates voltage spikes transmitted to other neurons (not shown) through memristor-based artificial synapses. The memristive $BiFeO_3$(BFO) can serve as an analog resistive switch (Shuai et al., 2011) with multiple distinguishable low resistance states (LRSs) (Shuai et al., 2013; Jin et al., 2014) and with a single detectable high resistance state (HRS). Due to the thermal diffusion of Ti atoms and their substitutional incorporation into the lower part of the $BiFeO_3$ (BFO) layer during BFO thin film growth on a Pt/Ti bottom electrode, the barrier at the Pt/Ti bottom electrode is flexible.

Earlier we have shown that STDP and triplet plasticity with learning windows on the millisecond time scale can be faithfully emulated on BFO-based artificial synapses by applying 60–80 pairings of pre- and post-synaptic spikes (Mayr et al., 2012; Cederström et al., 2013). In this work we investigate a significantly wider range of timescale configurability, ranging from 25 ms to 125 μs. To the best of our knowledge, this kind of timescale configurability has not been shown in memristive synapses before. We also examine the evolution of the induced memristive weight change over time and provide several power consumption figures. By increasing the programming voltage (HRS/LRS writing pulse amplitude), it is possible to decrease the switching pulse width as well as the power consumption during a single STDP writing process on BFO-based artificial synapses. Furthermore, the increased programming voltage also shortens the total pairing spike time, and enables to move from the standard biology-like 60–80 spike pairing STDP experiment to a single pairing STDP experiment that results in the same weight/memristance change.

Our work is structured as follows: In Section Materials and Methods, we describe the non-volatile resistive switching of BFO–based artificial synapses and introduce the single pairing STDP pulse sequence. In Section Results, we present the measured learning window, memory consolidation, and energy consumption of the single pairing STDP in BFO-based artificial synapses and discuss configurability, energy consumption, and retention of weight change in Section Discussion. The paper is summarized and an outlook is given in Section Summary and Outlook.

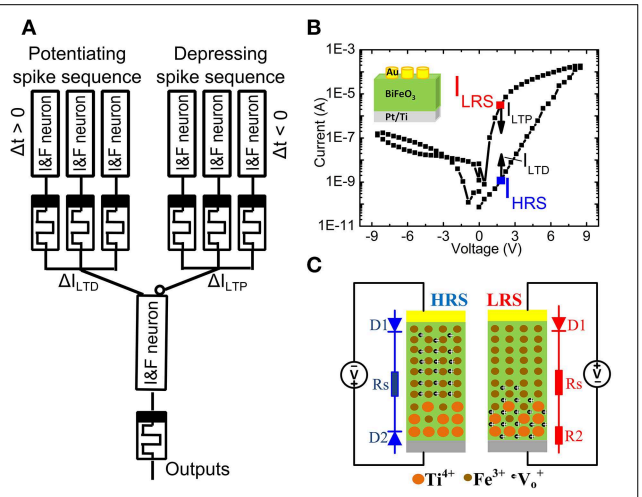


FIGURE 1 | (A) Schematic illustration of the memristor-based synaptic electronics. The artificial synapses are placed between Integrate & Fire neurons (I&F neuron). With a well-defined time delay Δt between the pre- and post-spikes the internal state ("weight") of the memristor is suitably changed. **(B)** Hysteretic current-voltage (IV) characteristics of a Au/$BiFeO_3$/Pt memristor in LRS and HRS with a top electrode area of 4.5E4 μm^2 under source voltages with maximum sweeping pulse amplitude of 8.5 V and a pulse width of 100 ms. The current in high resistance state I_{HRS} and in low resistance state I_{LRS} is read out at +2.0 V, after having switched the memristor into HRS and LRS, respectively. The long term potentiation current I_{LTP} and the long term depression current I_{LTD} lie below the reading current in LRS (I_{LRS}) and HRS (I_{HRS}). Inset shows the structure of a BFO memristor. **(C)** Schematic demonstration of the distribution of fixed Ti^{4+}, fixed Fe^{3+} and mobile V_O^+.

Materials and Methods

Nonvolatile, Analog Resistive Switching in $BiFeO_3$

Polycrystalline, 600 nm thick $BiFeO_3$ (BFO) thin films with a flexible bottom barrier have been grown by pulsed laser deposition on Pt/Ti/SiO_2/Si substrates. Circular Au top contacts have been magnetron sputtered on the BFO thin films using a shadow mask (Shuai et al., 2011, 2013; Jin et al., 2014). The Pt/Ti bottom electrode and the Au top contacts posses a flexible and a fixed barrier height, respectively. As illustrated in **Figure 1B**, by applying the sweeping source voltage from 0 V → −8.5 V → +8.5 V → 0 V between the Au top electrode and the bottom electrode, the current-voltage characteristics, which were recorded using a Keithley source meter 2400, reveal reproducible nonvolatile hysteretic bipolar resistive switching in BFO memristors with mobile donors (oxygen vacancies) and fixed donors (Ti donors). As illustrated in **Figure 1C** which has been adapted from Ref. (You et al., 2014), the physical mechanism underlying resistive switching in BFO memristors is related with the nonvolatile change of flexible barriers in Ti-containing BFO memristors. Due to voltage application of a LRS writing pulse, fixed Ti donors close to the bottom electrode can effectively trap mobile oxygen vacancies in BFO. The bottom electrode becomes non-rectifying and the BFO memristor is in LRS. On the other hand, when applying the HRS switching pulse, the mobile donors in BFO memristors are redistributed between the top and the bottom electrode. The bottom electrode becomes rectifying and the BFO memristor is in HRS. Note that for both writing pulses the Au top electrode remains rectifying.

A single writing pulse with an amplitude $V_w = +8.0$ V and −8.0 V can be used to switch the BFO memristor into LRS and HRS, respectively. The maximum possible amplitude increases with the thickness of the BFO memristor and decreases with

the length of the writing pulse. For a BFO layer thickness of 600 nm and a writing pulse length of 100 ms, the barrier height of the bottom electrode typically starts to change at a writing pulse of amplitude $V_w = +3.0$ V. Applying a dc voltage below +2.0 V to the BFO memristor does not change the barrier height of the bottom electrode, and the state of the BFO memristor does not change. Therefore, the +2.0 V dc voltage is defined as the reading bias for the 600 nm thick BFO memristor. The ratio between the resistance R_{HRS} in HRS and the resistance R_{LRS} in LRS amounts to $R_{HRS}/R_{LRS} = 2770$ (**Figure 1B**). For changing the synaptic weight the absolute value of the amplitude V_p of the pre-synaptic and post-synaptic spike has to be larger than the reading bias amplitude +2.0 V (Smerieri et al., 2008; Borghetti et al., 2009; Lai et al., 2009). In our previous work, we used a 500 nm thick BFO layer and an amplitude of 2.3 and 2.0 V for STDP with 60–80 pairings of pre- and post-synaptic spikes. In this work, we use a 600 nm thick BFO layer and an amplitude V_p of 3.0 V for STDP with single pairing of pre- and post-synaptic spikes. For the potentiating (depressing) spike sequence, the long term potentiation current I_{LTP} (long-term depression current I_{LTD}) decreases exponentially with decreased pulse amplitude in positive (negative) voltage range: $I_{LRS} > I_{LTP}$ ($I_{HRS} < I_{LTD}$).

The nonvolatile resistive switching of BFO was examined by a retention test (**Figure 2A**). A single writing pulse of $V_w = +8.0$ V and −8.0 V and a pulse width of $t_p = 100$ ms was used to switch the BFO memristor into LRS and HRS, respectively. The reading currents have been read out with a reading bias of $V_r = +2.0$ V and are defined as the current of HRS (I_{HRS}) and LRS (I_{LRS}). As shown in **Figure 2A** the BFO memristor exhibits degradation of the LRS within the testing time of 2 h. No significant change has been observed for HRS during the retention time of 5 h. This non-ideal retention motivated us to investigate memory consolidation (Clopath et al., 2008) in BFO with the shortened pulse sequence of single pairing STDP.

A BFO memristor with multilevel resistive switching can be considered as an analog resistive switch and used as an artificial synapses. The retention of multilevel resistive switching is illustrated in **Figure 2B**. Positive writing pulses ranging from 2.0 to 8.0 V are applied to the BFO-based artificial synapse. As expected from the current-voltage characteristics (**Figure 1B**), the reading current at 2.0 V increases with increasing amplitude of the writing bias. After applying the positive writing pulses V_w (as switched, $t_w = 2$ s), the reading current was largest and slightly decreased (30 mins, $t_w = 30$ min) with increasing waiting time t_w (**Figure 2B**). However, due to the degradation (**Figure 2B**) different LRSs will become indistinguishable. E.g., the reading current for a writing bias of $V_w = 5.5$ V and a waiting time of t_w= 2 s is the same as the reading current for $V_w = 6.0$ V and $t_w = 30$ min. We have already shown that the retention of BFO memristors can be significantly improved by an additional BFO surface modification using low energy Ar^+ ion irradiation before depositing the Au top electrode (Shuai et al., 2011). Optimized parameters for the Ar^+ irradiation process are discussed in Ref. (Ou et al., 2013). The Ar^+ irradiation helps to homogenize the average crystallite size in the polycrystalline BFO memristors.

Pulse Sequence for Single Pairing Spike-timing Dependent Plasticity

In our previous work, we have used a bias amplitude of $V_p = 2.3$ V for STDP with 60–80 pairings of pre- and post-synaptic spikes (Mayr et al., 2012; Cederström et al., 2013). Especially, Mayr et al. illustrates how the pre- and post-synaptic waveforms of a specific biology-derived synaptic plasticity rule (Mayr and Partzsch, 2010) can be adjusted to operate the BFO memristors. The resulting waveforms are comparable to the waveforms proposed by Zamarreño-Ramos et al. (2011). In order to shorten the total pairing spike time, in this work we slightly increased the bias amplitude to $V_p = 3.0$ V and applied a single pre- and

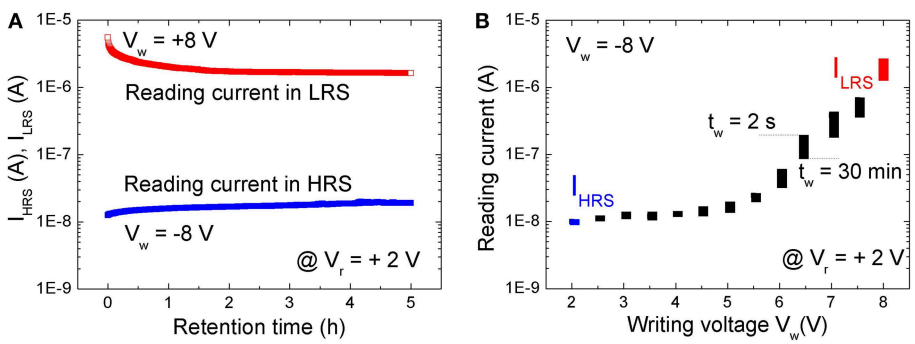


FIGURE 2 | (A) Retention test with a reading bias of $V_r = +2.0$ V after setting the BFO memristor to LRS (red symbols) and to HRS (blue symbols). The reading current has been recorded every 30 s. **(B)** Retention of multilevel resistive switching in a BFO memristor, which has been initially set to HRS by a writing voltage of $V_W = -8.0$ V. The reading current has been measured at a small reading bias of $V_r = +2.0$ V directly after switching BFO into one of the multiple LRSs with a positive writing bias of V_W ranging from +2.0 to +8.0 V (top edge of the rectangles, $t_W = 2$ s) and 30 min later (bottom edge of the rectangles, $t_W = 30$ min). Note that the reading current starts to increase for a writing voltage of ca. +3.0 V, i.e., the state of the BFO starts to change. All states in **(B)** are read with a pulsed reading bias amplitude of $V_r = +2.0$ V and length 100 ms. Because the reading current changes from $I_r = 1.1$E-2 μA in HRS with $R = 1.8$E8 Ω to $I_r = 2$ μA in LRS with $R = 1$E6 Ω, the power ($P = R \cdot I^2$) will change from 2.2E-8 W in HRS to 4.0E-6 W in LRS. The resolution of a pulsed power meter amounts to 0.01 dB. So theoretically more than 2000 power levels would be achievable, and we expect that at least 32/64 levels are possible in a power efficient manner.

post-synaptic spike. In comparison to what is discussed in Mayr et al. (2012), the single spike pairing instead of multiple (60–80) pairings allows us to shorten the total spike time and to adjust the learning time constant of the STDP function from 25 ms to 125 µs. The detailed signal scheme of Memristor initialization, single pairing STDP, and memory consolidation for long-term potentiation (LTP) and long-term depression (LTD) are shown in **Figure 3**. In order to facilitate reproducing this signal scheme, the parameters used in every step in the pulse sequence are listed in **Table 1**. As illustrated in **Figure 6A** the signal scheme for resistive switching from HRS into a single LRS (**Figure 6B**) can be simplified and reduced to Memristor initialization for

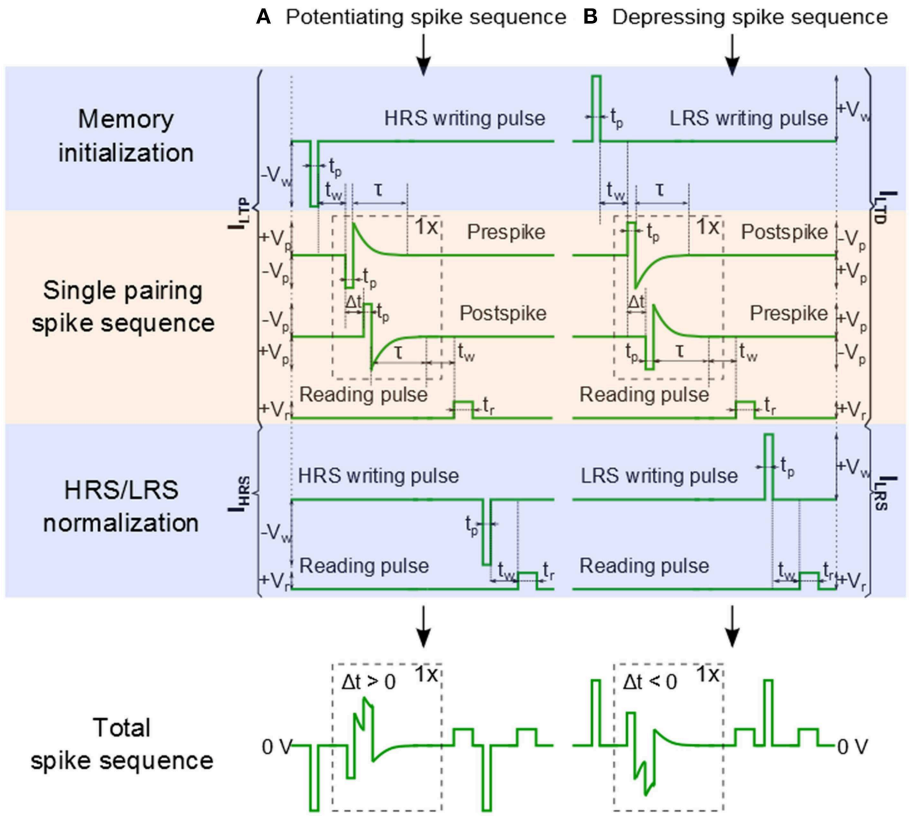


FIGURE 3 | Signal scheme of Memristor initialization, Single pairing STDP, and Memory consolidation. (A) A pre-post spike order is used for long term potentiation (LTP). **(B)** A post-pre spike order is used for long term depression (LTD). The potentiation current I_{LTP} (depression current I_{LTD})and the initial HRS current I_{HRS} (and the initial LRS current I_{LRS}) are used to normalize the long term potentiation current ΔI_{LTP} (the long term depression current ΔI_{LTD}) as defined in Equations (2) and (3). t_p is the pulse width and t_w is the measurement waiting time before applying the reading pulse V_r.

TABLE 1 | Parameters for the potentiating spike sequence (Δt > 0) and for the depressing spike sequence (Δt < 0) during Memristor initialization, Memory consolidation, and Single pairing STDP.

Step in pulse sequence	Memristor initialization	Memory consolidation	Single pairing STDP	Memory consolidation	Memory consolidation
Potentiating spike sequence	$-V_w$ & t_p	t_w	$-V_p$ & t_p/$+V_p$ & τ $\Delta t > 0$ $+V_p$ & t_p/$-V_p$ & τ	t_w	$+V_r$ & t_r
Depressing spike sequence	$+V_w$ & t_p	t_w	$+V_p$ & t_p/$-V_p$ & τ $\Delta t < 0$ $-V_p$ & t_p/$+V_p$ & τ	t_w	$+V_r$ & t_r

The amplitude $|V_w|$and the length t_p of the writing bias pulse determine the Memristor initialization. The waiting time t_w after Memristor initialization, the waiting time t_w after Single pairing STDP and the amplitude $|V_r|$ and the length t_r of the reading bias pulse determine Memory consolidation. The amplitude $-V_p$, the length t_p, the amplitude $+V_p$ and exponential decay τ determine the presynaptic spike and the amplitude $+V_p$, the length t_p, the amplitude $-V_p$, and the exponential decay τ determine the post-synaptic spike. The time delay between the pre- and the post-spike is defined by Δt.

LTP and to Memory consolidation for LTD (**Figure 6A**). The step labeled Memristor initialization refers to the application of a writing pulse to set the BFO memristor in HRS and LRS. In the HRS the BFO memristor has both rectifying top and bottom electrodes whereas in the LRS the BFO memristor has a rectifying top electrode and a non-rectifying bottom electrode (You et al., 2014). For the pulse order leading to potentiation (**Figure 3A**), a single negative pulse, i.e., the HRS writing pulse, is applied to switch the memristive device into HRS. After the waiting time t_w a single pre- and a single post-spike is applied to the top electrode of device. The pre- and post-spikes superimpose at the BFO memristor as potentiating spike, and the spike timing difference Δt determines the waveform of the potentiating spike ($\Delta t = t_p > 0$ for the potentiating inputs). Each pre- and post-spike consists of one rectangular pulse with pulse amplitude V_p and one exponentially decaying pulse V_{exp}

$$V_{exp} = |V_p| \bullet exp\left(\frac{-t}{\tau}\right), \tag{1}$$

with the decay time $\tau = \tau_{pre} = \tau_{post}$, where τ_{pre} and τ_{post} are the exponential decay times of pre- and post-spikes, respectively. In order to reduce the influence of the exponential decay on the single pairing STDP function, we choose $\tau = 2.5 \cdot t_p$. For the potentiating (depressing) spike order, the spike timing difference Δt between the pre- and post-spike is positive (negative) and lies in the range: $t_p = |\Delta t| = 10 \cdot t_p$. In both pre- and post-spikes, the rectangular pulse is short compared to the decay time of the exponential waveform, and the amplitude of the overlapped spike pulses depends on the spike time difference Δt between both waveforms. After the measurement waiting time t_w the synaptic weight of BFO-based artificial synapses has been checked by applying a reading bias of $V_r = +2.0$ V with a pulse width of $t_r = 100$ ms. The reading current is defined as the potentiation current I_{LTP} anddepression current I_{LTD} after sourcing potentiating spike and depressing spike, respectively.

Finally, the reading current I_{HRS} (I_{LRS}) of BFO in HRS (LRS) is measured at a reading bias of $V_r = +2.0$ V after recording I_{LTP} (I_{LTD}). For biological reasons it is desirable to keep STDP bounded. Therefore, we have normalized the LTP and LTD current values. After a potentiating spike sequence the synaptic weight scales with the normalized potentiation current ΔI_{LTP}

$$\Delta I_{LTP}\,(\%) = \frac{I_{LTP} - I_{HRS}}{I_{LTP}} * 100\%, \tag{2}$$

and after a depressing spike sequence the synaptic weight scales with the normalized depression current ΔI_{LTD}

$$\Delta I_{LTD}\,(\%) = \frac{I_{LTD} - I_{LRS}}{I_{LRS}} * 100\%. \tag{3}$$

After normalization using Equations (2) and (3) LTP lies in the range from 0 to +100% and LTD lies in the range from 0 to −100%, respectively. As we have shown in Mayr et al. (2012), the specific STDP characteristics can be configured through the waveform. Specifically, τ_{pre} directly translates to the STDP pre-post time window, while τ_{post} translates to the post-pre time window. The V_p of the pre- and post-pulses translate to the respective scaling of the STDP amplitudes.

Results

In the following single pairing STDP in BFO-based artificial synapses (Section Nonvolatile, Analog Resistive Switching in BiFeO3) is demonstrated by using different pulse widths t_p and measurement waiting times t_w. The potentiating and depressing input signals (Section Pulse Sequence for Single Pairing Spike-timing Dependent Plasticity) have been generated with an Agilent pulse function arbitrary generator 81150A. The reading current has been measured with a Keithley 2400 source meter.

Learning Window

According to the input signal scheme (**Figure 3**) the BFO memristor is set in the HRS and in the LRS with a writing pulse amplitude of $V_w = -8.0$ and +8.0 V, respectively. For the single pairing STDP measurements on a BFO-based artificial synapse pre- and post-spikes of different pulse widths $t_p = 10$ ms, 1 ms, 500 µs, and 50 µs, and with a pulse amplitude of $|\pm V_p| = 3.0$ V, and a waiting time t_w 10 s have been chosen (**Figure 4**). The exponential decay time constant ($\tau = 2.5 \cdot t_p$) amounts to $\tau = 25$ ms (**Figure 4A**), 2.5 ms (**Figure 4B**), 1.25 ms (**Figure 4C**), and 125 µs (**Figure 4D**). After recording I_{LTP} (I_{LTD}) the reading current I_{HRS} (I_{LRS}) of BFO in HRS (LRS) has been measured at a reading bias of $V_r = +2.0$ V and the normalized potentiation current ΔI_{LTP} Equation (2) and the normalized depression current ΔI_{LTD} Equation (3) are calculated. The synaptic weight of the BFO memristor scales with the normalized potentiation current ΔI_{LTP} and the normalized depression current ΔI_{LTD}. If the prespike precedes the post-spike ($\Delta t > 0$) biological synapses (Bi and Poo, 1998) undergo long term potentiation LTP, i.e., the connection between two neurons becomes stronger. On the other hand, if the post-spike precedes the prespike ($\Delta t < 0$), biological synapses undergo long term depression LTD, i.e., the connection between two neurons becomes weaker. We have measured the LTD current I_{LTD} and the LTP current I_{LTP} in a BFO-based artificial synapse and can show that the BFO memristor emulates the STDP function of biological synapses. The normalized current ΔI decreases with increasing delay time $|\Delta t|$. The normalized current curve for positive and negative Δt is the LTP and LTD curve (**Figure 4**), respectively. As an example, in the following we discuss the LTP curve in **Figure 4** for $\Delta t = t_p > 0$. Initially the BFO-based artificial synapse is set into HRS. The maximum amplitude of the potentiating spike amounts to $2V_p = +6.0$ V. For this potentiating spike the BFO-based artificial synapse is fully switched to LRS. The normalized potentiation current ΔI_{LTP} at $\Delta t = t_p$ amounts to ca. 100%. In the time delay range $0 < t_p < \Delta t \leq 10 \cdot t_p$, the maximum amplitude of potentiating spikes is reduced from 6.0 to 3.2 V. Therefore, the exponential-like decay of the normalized current dominates STDP with increasing Δt and the synapse cannot be fully switched to LRS by applying these potentiating spikes. For both positive and negative time delays $|\Delta t| = 10 \cdot t_p$, ΔI decreases

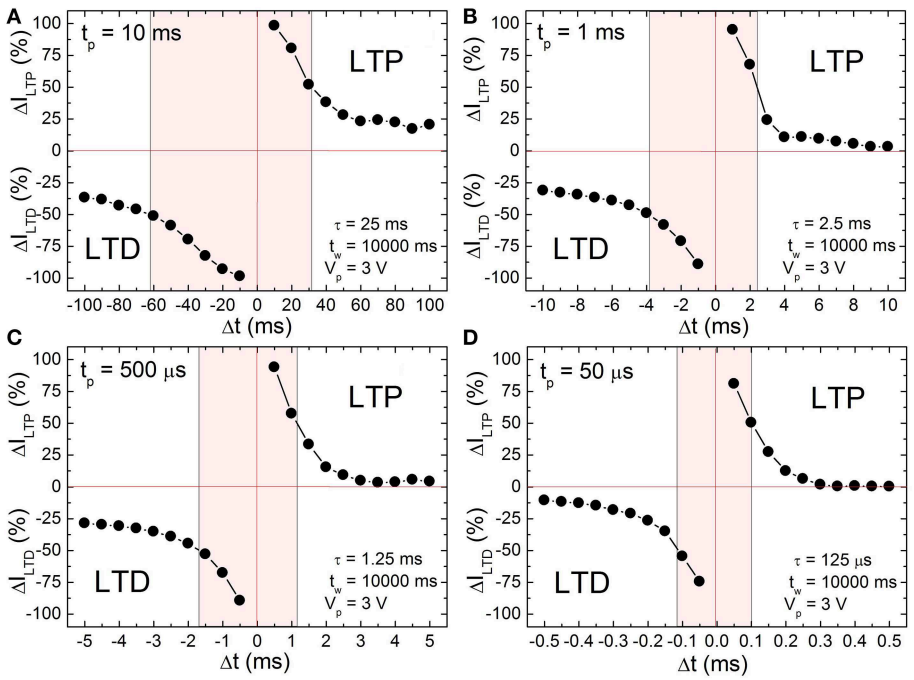


FIGURE 4 | Long term depression current ΔI_{LTD} (negative range of y-axis) and long term potentiation current ΔI_{LTP} (positive range of y-axis) of a ca. 600 nm thick BFO memristor with a contact area of 4.5E4 μm^2 for single pairing STDP with pulse width (A) t_p = 10 ms, (B) t_p = 1 ms, (C) t_p = 500 μs and (D) t_p = 50 μs, measurement waiting time t_w = 10000 ms, pulse amplitude V_p = 3.0 V, reading pulse amplitude V_r = +2.0 V and reading pulse width t_r = 100 ms. ΔI_{LTD} and ΔI_{LTP} have been normalized using Equations (2) and (3), respectively. The memristor was preset in HRS and LRS (Memristor initialization in **Table 1**) with a writing pulse amplitude of $V_W = -8.0$ V and $V_W = +8.0$ V, respectively.

with decreasing pulse width t_p. At t_p = 500 μs and 50 μs, ΔI_{LTP} amounts to 0% at $|\Delta t| = 10 \cdot t_p$. It is also noticed that ΔI_{LTP} decreases more strongly than ΔI_{LTD} in the larger time delay range. That is because the threshold voltage for LRS is higher than the threshold voltage for HRS. For example in Ref. (Mayr et al., 2012) a voltage of 2.3 V and of 2.0 V has been used as the threshold voltage to switch a BFO-based artificial synapse to LRS and HRS, respectively. The shaded regions in **Figure 4** show the ranges of the delay time Δt where the normalized current is larger than 50% for four different pulse widths t_p. This range is also called learning window and decreases from 25 ms to 125 μs with decreasing pulse width t_p from 10 ms to 50 μs.

As can be seen from **Figure 4**, the STDP time windows can be finely controlled. Specifically, making Δt longer results in a monotonous decrease in both potentiation and depression with increasing Δt, i.e., the memristance change directly and fine grainedly follows the applied waveform resulting from the overlay of pre- and post-pulse. This is in contrast to most other reported memristive synapses, where the time difference between pre- and post-pulse only translates to a stochastic, average change of memristance (Jo et al., 2010; Alibart et al., 2012).

Memory Consolidation

Memory consolidation has been investigated in models of biology in order to improve the understanding of the translation of an initially induced weight change to long term weight stabilization (Anokhin, 2005; Clopath et al., 2008). This motivated us to investigate the memristance weight, i.e., memory consolidation, in BFO-based artificial synapses in more details by performing single pairing STDP measurements with different waiting times t_w (2 s = t_w = 5 h). In biological systems, the waiting time corresponds to the time which elapses before something learned is retrieved. On the other hand, for the memory consolidation measurements, we have again used the ca. 600 nm thick BFO-based artificial synapses and applied a writing voltage of $V_w = +6.0$ V. In **Figure 5A** the corresponding STDP data are plotted for t_w = 2, 60, and 300 s. We have chosen single pre- and post-synaptic spikes with the same absolute value of the pulse amplitude V_p = 3.0 V, pulse width t_p = 10 ms and exponential decay time τ = 25 ms. As shown in **Figure 5A**, the LTP and LTD curves shift toward low normalized current values with increasing waiting time in both positive and negative spike timing ranges. Therefore, the dependence of LTP and LTD on the writing pulse amplitude can be used to trace differences in the LTP and LTD curves of single pairing STDP. For BFO-based artificial synapses with a smaller writing voltage V_w, the optimized STDP curve with more significant exponential-like function (as shown in **Figure 4**) is reproducible by choosing a smaller pulse amplitude V_p, e.g., V_p = 2.5 V.

Furthermore, memory consolidation measurements (**Figure 5B**) reveal that for a waiting time t_w shorter than 1 h there is a visible change of reading current (degradation)

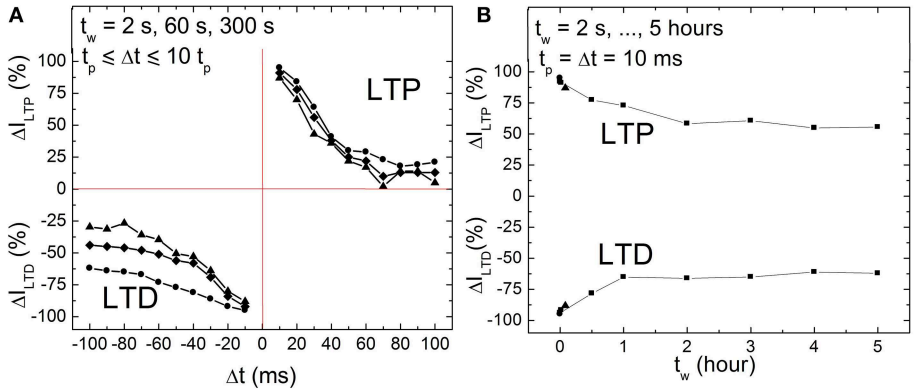


FIGURE 5 | (A) STDP of a BFO-based artificial synapses with different waiting times $t_W = 2\,s$ (circles), 1 min (quadrangles), and 5 min (triangles) for $t_P = \Delta t = 10t_P$. Pulse amplitude $V_P = 3.0\,V$, pulse width $t_P = 10\,ms$, and exponential decay time $\tau = 25\,ms$. **(B)** Memristance weight consolidation for a fixed $\Delta t = t_P = 10\,ms$ and for a waiting time of $t_W = 2\,s$ (circles), 60 s (quadrangles), and 300 s (triangles) from **(A)** and $t_W = 0.5,\ 1,\ 2,\ 3,\ 4,\ 5\,h$ (squares). The pulse amplitude V_P amounts to 3.0 V. The exponential decay amounts to $\tau = 25\,ms$. The writing voltage for Memristor initialization amounts to $|\pm\ V_W| = 6.0\,V$.

both in positive and negative spike timing ranges after applying a single pre-synaptic and post-synaptic pulse sequence, whereas for a waiting time t_w longer than 2 h the current is stabilized. This is in agreement with the results from retention measurements (**Figure 2A**).

Energy Consumption

Low energy efficiency, large chip size, and complex STDP synapse circuits are major bottlenecks of today's bio-inspired systems, e.g., neural networks where synapses typically outnumber neurons by more than 500:1. In order to reliably observe STDP functionality the corresponding current changes should lie in the nA current range and above. In addition to the stabilization of multilevel resistive switching, we can also increase the current level in a controlled manner by low-energy Ar^+ ion irradiation (Ou et al., 2013). This will allow for integrating BFO-based artificial synapses with smaller contact area A (**Table 2**), e.g., in neural networks, without adding another device for amplifying current changes. The estimated energy consumption of each synapse in human brain amounts to only 1–10 fJ (**Table 2**). In order to approach the high energy efficiency of biological synapses, we applied single pairing (not 60–80 pairing) STDP pulses to BFO-based artificial synapses. For single pairing STDP most of the energy is consumed during SET operation, e.g., Memristor initialization into LRS (**Table 1**, **Figure 3**). For example, in $TiN/Ge_2Sb_2Te_5/TiN/W$ artificial synapses the energy for SET operation is 50 pJ while the energy for RESET operation is 0.675 pJ Ref. (Kuzum et al., 2012).

The energy consumed during SET operation is

$$E = V'_w \bullet I_{avg} \bullet t'_p, \tag{4}$$

with $I_{avg} = I_{peak}/2$. The writing voltage amplitude V_w, the setting current I_{peak}, and writing pulse width t_p are the crucial parameters for evaluating the energy consumption. Note that for the polycrystalline BFO memristors with different sizes of BFO crystallites, larger BFO crystallites below the top electrode are possibly not switchable. Therefore, the effective area of the top electrode might be smaller than the nominal area of the top electrode. Using BFO-based artificial synapses we can downscale the size of the top electrodes (Jin et al., 2014), increase the pulse amplitude V'_w and also reduce the pulse width t'_p Equation (4) to further decrease the energy consumption per setting process (**Figure 6**).

In order to optimize the energy efficiency of BFO-based artificial synapses, we have applied a large writing pulse amplitude of 23.0 V to compensate the short pulse width of 50 ns. The corresponding energy consumption amounts to 4.7 pJ. The LRS reading current and HRS reading current at 2.0 V amount to 980 and 64 nA, respectively. The theoretical maximum normalized current ranges from 93.5 to 0% and from 0 to 93.5% in both curves Equation (2) and (3).

In **Table 2** (Kandel and Schwartz, 1985; Jo et al., 2010; Chang et al., 2011; Yu et al., 2011; Kuzum et al., 2012; Wu et al., 2012) different memristor-based artificial synapses are listed and compared with respect to their energy consumption per (re)setting process. The $TiN/Ti/AlO_x/TiN/Ti$ memristor (Wu et al., 2012) shows the smallest energy consumption of 1.5 pJ per SET pulse. It is expected that to a certain extent the energy consumption can be further reduced by further reducing the electrode area size A. However, one has to consider that BFO is a polycrystalline thin film and that only 1–0.1% of the crystallites below the top electrode of the polycrystalline BFO are switched in single pairing STDP.

Discussion

Configurability

In this work single pairing STDP in BFO-based artificial synapses has been demonstrated for emulating the functionality and the plasticity of biological synapses. The waveform-defined plasticity of BFO memristors in addition to their multilevel memristive

TABLE 2 | Energy consumption E, setting potential amplitude V_W, average setting current I_{avg}, pulse width t_p and top electrode area size A of resistive switching during SET operation of different memristor-based artificial synapses (Kandel and Schwartz, 1985; Jo et al., 2010; Chang et al., 2011; Yu et al., 2011; Kuzum et al., 2012; Wu et al., 2012).

Single synapse	E (pJ)	V_W (V)	I_{avg} (μA)	t_p (ns)	A (μm^2)
Human brain (total number of synapses $N = 10^{15}$, $P_{total} = 10$ W) (Kandel and Schwartz, 1985; Da Costa, 2013)	(1–10) *1E–3	-	-	-	0.12
TiN/Ti/AlO_x/TiN/Ti (Wu et al., 2012)	1.5	+1.5	+100	10	0.72
Au/BFO/Pt/Ti (this paper)	4.7	+23.0	+4.1	50	4.5E+4
TiN/HfO_x/AlO_x/Pt (Yu et al., 2011)	6.0	−2.5	−240	10	0.0079
TiN/$Ge_2Sb_2Te_5$/TiN/W (Kuzum et al., 2012)	50	−5.5	−900	10	0.018
CMOS-electrode/Ag + Si/CMOS-electrode (Jo and Lu, 2008)	430	+3.2	+0.45	3.0E+5	0.031
Pd/WO_x/W/SiO_2/Si (Chang et al., 2011)	520	+1.3	+0.40	1.0E+6	0.053

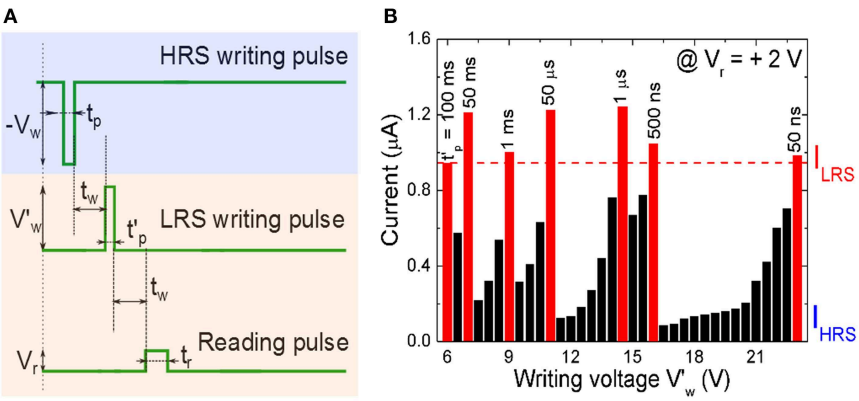


FIGURE 6 | (A) Signal scheme for resistive switching a BFO memristor in HRS into LRS. The memristor is initialized into the HRS by applying a writing voltage $V_W = -6.0$ V with a pulse width $t_p = 100$ ms, and is then switched back to different LRSs with different pulse amplitudes V'_W and pulse widths t'_p. **(B)** Reading current of the BFO memristor with a contact area of 4.5E4 μm^2 in LRS in dependence on the writing voltage V'_W in the range from 6.0 to 23.0 V and with different constant pulse widths of $t'_p = 50$ ms, 1 ms, 50 μs, 1 μs, 500 ns, and 50 ns. The reading voltage amounts to +2.0 V. For a given pulse width at least one writing voltage (red bar) is large enough to set the BFO memristor in the LRS. In that case the reading currents is even larger than the current I_{LRS} read out after applying a writing voltage of $V_W = +6.0$ V with a pulse width of $t'_p = 100$ ms (first red bar).

programming capability enables easy control of the STDP time windows, as evidenced by the three orders of magnitude timescale configurability shown in this paper. While there has been a lot of simulation work on this topic, the number of devices where STDP or variations have actually been implemented and measured is still fairly small (Jo et al., 2010; Alibart et al., 2012). Among those, our highly-configurable, finely grained learning curves are unique, other implementations exhibit statistical variations (Jo et al., 2010), can only assume a few discrete levels (Alibart et al., 2012) or the learning windows are device-inherent, i.e., cannot be adjusted (Ohno et al., 2011). We expect that for BFO-based artificial synapses at least 32/64 levels are possible in a power efficient manner. In addition, the wide range of timescales possible in BFO-based synapses enables e.g., a timebase-tunable system that could learn a classification offline in an accelerated manner, while still able to interact with real-time sensors before or after this learning.

As mentioned in the introduction, BFO-based artificial synapses can be used for conventional STDP experiments, where only multiple spike pairings exhibit significant weight change, as well as in the mode used in this paper, where a single pairing already induces a significant weight change. By changing the voltage of the pre- and post-synaptic pulses, any point in between these two extremes can also be chosen, again showing the excellent configurability of BFO-based artificial synapses. However, the versatility of BFO memristors comes at the price that in contrast to e.g., phase-change materials, BFO is not easily integrated on top of CMOS (Shuai et al., 2013).

Energy Consumption

In **Table 2**, we have shown an energy consumption of $E = 4.7$ pJ in a BFO-based artificial synapse with electrode size of 4.52E4 μm^2. While this is still three orders of magnitude above the energy consumption of biological synapses, it is one of the lowest reported so far for other artificial synapses. Compared to neuromorphic approaches, all memristive approaches are several orders of magnitude better (Azghadi et al., 2014). In terms of absolute area, the BFO memristor is comparable to some neuromorphic implementations (Hasler and Marr, 2013; Noack et al., 2015), but not competitive with memristor crossbar devices,

as we are employing a single device test structure that has a large contact size for reasons of convenience. However, BFO device scaling is well established, thus we can aggressively scale the size of the top electrode to $10\,\mu m^2$ and the thickness of the BFO to 100 nm (Jin et al., 2014). For BFO with larger electrode area size, the current scales linearly with area size. For smaller electrode area size we would expect that the current scales with the number of BFO crystallites below the electrode. And in the limit case of nanoscale electrodes, the smallest possible current should be the current through single BFO crystallites.

Retention of Weight Change

We have investigated the retention of memristance weight change across time. As **Figure 5A** shows, the basic shape of the STDP curves is preserved across time. **Figure 5B** illustrates that even after memory consolidation, we retain a graded weight, i.e., a unimodal weight distribution. Our synapse does not collapse in either a potentiated or depressed (bimodal) distribution as predicted in some synaptic models (Fusi et al., 2000; Clopath et al., 2008). In memristive literature, there is usually no investigation of these phenomena, the weight change is taken at some unspecified time after induction and then assumed to be non-volatile. Only very few articles have investigated the actual non-volatility/weight retention across time and shown that the assumption of a non-volatile change is not necessarily valid (Chang et al., 2011). Thus, compared to other reports, this article gives a neuromorphic designer a clear guide on how to utilize the memristive synapses for long-term storage.

Interestingly, this investigation of memory consolidation is also somewhat missing in the original biological measurements. Usually, data on the weight evolution ca. 30–60 min after induction is provided, but only on single example pairing experiments. These data points show various behaviors, from unchanged weights after initial weight induction (Froemke and Dan, 2002) to increases of weight change across time (Bi and Poo, 1998), decreases across time (Markram et al., 1997) or slow oscillations around the initial potentiated/depressed weight value (Sjöström et al., 2001). However, it is unclear how the overall STDP window consolidates over time. Thus, measuring the evolution of an STDP curve across time after induction at biological synapses similar to our investigation on memristive synapses may actually be a quite interesting scientific question.

Summary and outlook

In this work we have investigated a wide range of timescale configurability, ranging from 25 ms to 125 µs. Also, we have investigated power consumption figures and have shown that it is possible to decrease the switching pulse width and to reduce the power consumption during a single STDP writing process on BFO-based artificial synapses to only 4.5 pJ. Furthermore, the increased programming voltage also shortens the total pairing spike time, and enables to move from the standard biology-like 60–80 spike pairing STDP experiment to a single pairing STDP experiment with the same weight/memristance change.

One important advantage of single STDP in comparison to 60–80 spike STDP is that both pre- and post-synaptic waveform are causal, i.e., they start only at the pre- respectively post-synaptic pulse. This is in contrast to most currently proposed waveforms for memristive learning, where the waveforms have to start well in advance of the actual pulse (Zamarreño-Ramos et al., 2011), which requires pre-knowledge of a pulse occurrence. Especially, in an unsupervised learning context with self-driven neuron spiking, this pre-knowledge is simply not existent.

In a wider neuroscience context, waveform defined plasticity as shown here could be seen as a general computational principle, i.e., synapses are not likely to measure time differences as in native forms of STDP rules, they are more likely to react to local static (Ngezahayo et al., 2000) and dynamic (Dudek and Bear, 1992) state variables. In the future some interesting predictions could be derived from that, e.g., STDP time constants that are linked to synaptic conductance changes or to the membrane time constant (Pfister et al., 2006; Mayr and Partzsch, 2010). These predictions could be easily verified experimentally.

Acknowledgments

ND acknowledges funding by BMWi-ZIM (VP2999601ZG2). CM acknowledges funding by the European Union Seventh Framework Programme (FP7/2007- 2013) under grant agreement no. 269459 (CORONET) and no. 612058 (RAMP). HS and DB are grateful for financial support from the Deutsche Forschungsgemeinschaft (SCHM 1663/4-1,2, BU 2956/1-1) and the Networking Fund of the Helmholtz Association (VH-VI-422).

References

Adee, D. (2009). *IBM Unveils a New Brain Simulator. IEEE Spectr.* Available online at: http://spectrum.ieee.org/computing/hardware/ibm-unveils-a-new-brain-simulator

Alibart, F., Pleutin, S., Bichler, O., Gamrat, C., Serrano-Gotarredona, T., Linares-Barranco, B., et al. (2012). A memristive nanoparticle/organic hybrid synapstor for neuroinspired computing. *Adv. Funct. Mater.* 22, 609–616. doi: 10.1002/adfm.201101935

Ananthanarayanan, R., Esser, S. K., Simon, H. D., and Modha, D. S. (2009). "The cat is out of the bag: cortical simulations with 10^9 neurons and 10^{13} synapses," in *Proceedings IEEE/ACM Conference High Performance Networking Computing* (Portland, OR: IEEE), 1–12. doi: 10.1145/1654059.1654124

Anokhin, K. V. (2005). "Memory consolidation: narrowing the gap between systems and molecular genetics neurosciences," in *Complex Brain Functions: Conceptual Advances in Russian Neuroscience,* eds R. Miller, A. M. Ivanitsky, and P. M. Balaban (Amsterdam, FL: CRC Press), 51–71.

Azghadi, M. R., Iannella, N., Al-Sarawi, S., Indiveri, G., and Abbott, D. (2014), Spike-based synaptic plasticity in silicon: design, implementation, application, and challenges. *Proc. IEEE*, 102, 717–737. doi: 10.1109/JPROC.2014.2314454

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

Borghetti, J., Li, Z., Straznicky, J., Li, X., Ohlberg, D. A. A., Wu, W., et al. (2009). A hybrid nanomemristor/transistor logic circuit capable of self-programming. *Proc. Nat. Acad. Sci. U.S.A.* 106, 1699–1703. doi: 10.1073/pnas.0806642106

Cederström, L., Starke, P.,Mayr, C.,Shuai, Y., Schmidt, H., and Schüffny, R. (2013). "A model based comparison of $BiFeO_3$ device applicability in neuromorphic hardware," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)* (Beijing: IEEE), 2323–2326. doi: 10.1109/ISCAS.2013.6572343

Chang, T., Jo, S. H., and Lu, W. (2011). Short-term memory to long-term memory transition in a nanoscale memristor. *ACS Nano* 5, 7669–7676. doi: 10.1021/nn202983n

Chua, L. O. (1971). Memristor – The missing circuit element. *IEEE Transact. Circuit Theory* 18, 507–519.

Clopath, C., Ziegler, L., Vasilaki, E., Büsing, L., and Gerstner, W. (2008). Tag-trigger-consolidation: a model of early and late long-term-potentiation and depression. *PLoS Comput. Biol.* 4:e1000248. doi: 10.1371/journal.pcbi.1000248

Da Costa, N. M. (2013). Diversity of thalamorecipient spine morphology in cat visual cortex and its implication for synaptic plasticity. *J. Comp. Neurol.* 521, 2058–2066. doi: 10.1002/cne.23272

Di Lorenzo, P. M., and Victor, J. D. (2013). Spike *T*iming: *M*echanisms and *F*unctions, *Front. Neurosci*. Boca Raton, FL: CRC Press.

Du, N., Shuai, Y., Luo, W. B., Mayr, C., Schüffny, R., Schmidt, O. G., et al. (2013). Practical guide for validated memristance measurements. *Rev. Sci. Instrum.* 84:023903. doi: 10.1063/1.4775718

Dudek, S., and Bear, M. (1992). Homosynaptic long-term depression in area CAl of hippocampus and effects of N-methyl-D-aspartate receptor blockade. *Proc. Natl. Acad. Sci. U.S.A.* 89, 4363–4367. doi: 10.1073/pnas.89.10.4363

Froemke, R. C., and Dan, Y. (2002). Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature* 416, 433–438. doi: 10.1038/416433a

Fusi, S., Annunziato, M., Badoni, D., Salamon, A., and Amit, D. J. (2000). Spike-driven synaptic plasticity: theory, simulation, VLSI implementation. *Neural Comput.* 12, 2227–2258. doi: 10.1162/089976600300014917

Hasler, J., and Marr, B. (2013). Finding a roadmap to achieve large neuromorphic 593 hardware systems, *Front. Neurosci.* 7:118. doi: 10.3389/fnins.2013.00118

Indiveri, G., Chicca, E., and Douglas, R. (2006). A VLSI array of low power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Trans. Neural. Netw.* 17, 211–221. doi: 10.1109/TNN.2005.860850

Jin, L., Shuai, Y., Ou, X., Siles, P. F., Zeng, H. Z., You, T., et al. (2014). Resistive switching in unstructured, polycrystalline $BiFeO_3$ thin films with downscaled electrodes. *Phys. Status Solidi. A* 211, 2563–2568. doi: 10.1002/pssa.201431298

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano. Lett.* 10, 1297–1301. doi: 10.1021/nl904092h

Jo, S. H., and Lu, W. (2008). CMOS compatible nanoscale nonvolatile resistance switching memory. *Nano Lett.* 8, 392–397. doi: 10.1021/nl073225h

Kandel, E. R., and Schwartz, J. H. (1985). *Principles of Neural Science, 2nd Edn.* New York; Amsterdam; Oxford: Elsevier.

Kuzum, D., Jeyasingh, R. G. D., Lee, B., and Wong, H.-S. P. (2012). Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Lett.* 12, 2179–2186. doi: 10.1021/nl201040y

Koickal, T. J., Hamilton, A., Pearce, T. C., Tan, S. L., Covington, J. A., and Gardner, J. W. (2006). "Analog VLSI design of an adaptive neuromorphic chip for olfactory systems," in *Proceedings: IEEE International Symposium on: Circuits and Systems 2006. ISCAS 2006* (New York, NY: IEEE). doi: 10.1109/TCSI.2006.888677

Lai, Q., Zhang, L., Li, Z., Stickle, W. F., Williams, R. S., and Chen, Y. (2009). Analog memory capacitor based on field-configurable ion-doped polymers. *Appl. Phys. Lett.* 95, 213503. doi: 10.1063/1.3268433

Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* 275, 213–215. doi: 10.1126/science.275.5297.213

Mayr, C., Stärke, P., Partzsch, J., Cederstroem, L., Schüffny, R., Shuai, Y., et al. (2012). Waveform driven plasticity in $BiFeO_3$ memristive devices: model and implementation. *Adv. Neural Inf. Process. Syst.* 25, 1700–1708. Available online at: http://papers.nips.cc/paper/4595-waveform-driven-plasticity-in-bifeo3-memristive-devices-model-and-implementation.pdf

Mayr, C. G., and Partzsch, J. (2010). Rate and pulse based plasticity governed by local synaptic state variables. *Front. Synaptic. Neurosci.* 2:33. doi: 10.3389/fnsyn.2010.00033

Mayr, C., Partzsch, J., Noack, M., and Schüffny, R. (2013), "Live demonstration: multiple-timescale plasticity in a neuromorphic system," in *IEEE International Symposium on Circuits and Systems ISCAS 2013* (Beijing: IEEE), 666–670. doi: 10.1109/ISCAS.2013.6571933

Ngezahayo, A., Schachner, M., and Artola, A. (2000). Synaptic activity modulates the induction of bidirectional synaptic changes in adult mouse hippocampus. *J. Neurosci.* 20, 2451–2458. Available online at: http://www.jneurosci.org/content/20/7/2451.abstract

Noack, M., Partzsch, J., Mayr, C., Hänzsche, S., Scholze, S., Höppner, S., et al. (2015). Switched-capacitor realization of presynaptic short-term plasticity and stop-learning synapses in 28 nm CMOS. *Front. Neurosci.* 9:10. doi: 10.3389/fnins.2015.00010

Ohno, T., Hasegawa, T., Tsuruoka, T., Terabe, K., Gimzewski, J., and Aono, M. (2011), Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. *Nat. Mater.* 10, 591–595. doi: 10.1038/nmat3054

Ou, X., Shuai, Y., Luo, W., Siles, P. S., Kögler, R., Fiedler, J., et al. (2013). Forming-free resistive switching in multiferroic $BiFeO_3$ thin films with enhanced nanoscale shunts. *ACS Appl. Mater. Interfaces* 5, 12764–12771. doi: 10.1021/am404144c

Pfister, J.-P., Toyoizumi, T., Barber, D., and Gerstner, W. (2006). Optimal spike-timing dependent plasticity for precise action potential firing in supervised learning. *Neural Comput.* 18, 1309–1339. doi: 10.1162/neco.2006.18.6.1318

Schemmel, J., Grübl, A., Hartmann, S., Kononov, A., Mayr, C., Meier, K., et al. (2012). "Live demonstration: a scaled-down version of the BrainScaleS wafer-scale neuromorphic system," in *IEEE International Symposium on Circuits and Systems ISCAS 2012* (Seoul: IEEE), 702. doi: 10.1109/ISCAS.2012.6272131

Seo, K., Kim, I., Jung, S., Jo, M., Park, S., Park, J., et al. (2011). Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device. *Nanotechnology* 22:254023. doi: 10.1088/0957-4484/22/25/254023

Shuai, Y., Ou, X., Luo, W. B., Du, N., Wu, C., Zhang, W., et al. (2013). Nonvolatile multilevel resistive switching in Ar^+ irradiated $BiFeO_3$ thin films. *IEEE Electron Device Lett.* 34, 54–56. doi: 10.1109/LED.2012.2227666

Shuai, Y., Zhou, S. Q., Bürger, D., Helm, M., and Schmidt, H. (2011). Nonvolatile bipolar resistive switching in $Au/BiFeO_3/Pt$. *J. Appl. Phys.* 109, 124117. doi: 10.1063/1.3601113

Sjöström, P. J., Turrigiano, G. G., and Nelson, S. B. (2001). Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* 32, 1149–1164. doi: 10.1016/S0896-6273(01)00542-6

Smerieri, A., Berzina, T., Erokhin, V., and Fontana, M. P. (2008). Polymeric electrochemical element for adaptive networks: pulse mode. *J. Appl. Phys.* 104, 114513. doi: 10.1063/1.3033399

Snider, G. S. (2008). "Spike-timing-dependent learning in memristive nanodevices," in *Proceedings IEEE International Symposium Nanoscale Architechtures* (Anaheim, CA: IEEE), 85–92. doi :10.1109/NANOARCH.2008.4585796

Thomas, A., and Kaltschmidt, C. (2014). Elektronische Nervenzellen. *Phys. Unserer. Zeit.* 45, 21–25. doi: 10.1002/piuz.201301344

Wu, Y., Yu, S., Wong, H.-S. P., Chen, Y. S., Lee, H.-Y., Wang, S.-M., et al. (2012). "AlO_x-based resistive switching device with gradual resistance modulation for neuromorphic device application," in Memory Workshop (IMW), 2012 *4th IEEE International* (Milan: IEEE), 1–4. doi: 10.1109/IMW.2012.6213663

You, T., Du, N., Slesazeck, S., Mikolajick, T., Li, G., Bürger, D., et al. (2014). Bipolar electric-field enhanced trapping and detrapping of mobile donors in $BiFeO_3$ memristors. *ACS Appl. Mater. Interfaces* 6, 19758–19765. doi: 10.1021/am504871g

Yu, S., Wu, Y., Jeyasingh, R., Kuzum, D., and Wong, H.-S. P. (2011). An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. *IEEE Trans. Electron Devices* 58, 2729–2737. doi: 10.1109/TED.2011.2147791

Zamarreño-Ramos, C., Camuñas-Mesa, L. A., Pérez-Carrasco, J. A., Masquelier, T., Serrano-Gotarredona, T., and Linares-Barranco, B. (2011). On spike timing dependent plasticity, memristive devices, and building a self-learning visual cortex. *Front. Neurosci.* 5:26. doi: 10.3389/fnins.2011.00026

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in
NEUROSCIENCE

ORIGINAL RESEARCH ARTICLE
published: 22 July 2014
doi: 10.3389/fnins.2014.00201

Configurable analog-digital conversion using the neural engineering framework

Christian G. Mayr[1]*, Johannes Partzsch[2], Marko Noack[2] and Rene Schüffny[2]

[1] *Neuromorphic Cognitive Systems Group, Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland*
[2] *Electrical Engineering and Information Science, Chair of Highly Parallel VLSI Systems and Neuromorphic Circuits, Technische Universität Dresden, Dresden, Germany*

Edited by:
Julius Georgiou, University of Cyprus, Cyprus

Reviewed by:
Costa M. Colbert, Smart Logic, Inc., USA
Pantelis Georgiou, Imperial College London, UK

****Correspondence:***
Christian G. Mayr, Neuromorphic Cognitive Systems Group, Institute of Neuroinformatics, University of Zurich and ETH Zurich, Winterthurerstrasse 190, Zurich, Switzerland
e-mail: cmayr@ini.uzh.ch

Efficient Analog-Digital Converters (ADC) are one of the mainstays of mixed-signal integrated circuit design. Besides the conventional ADCs used in mainstream ICs, there have been various attempts in the past to utilize neuromorphic networks to accomplish an efficient crossing between analog and digital domains, i.e., to build neurally inspired ADCs. Generally, these have suffered from the same problems as conventional ADCs, that is they require high-precision, handcrafted analog circuits and are thus not technology portable. In this paper, we present an ADC based on the Neural Engineering Framework (NEF). It carries out a large fraction of the overall ADC process in the digital domain, i.e., it is easily portable across technologies. The analog-digital conversion takes full advantage of the high degree of parallelism inherent in neuromorphic networks, making for a very scalable ADC. In addition, it has a number of features not commonly found in conventional ADCs, such as a runtime reconfigurability of the ADC sampling rate, resolution and transfer characteristic.

Keywords: neural network analog digital converter, neural engineering framework, ADC with signal processing, multiple input ADC

1. INTRODUCTION

Circuits for analog-digital-conversion (ADC) are at the heart of every integrated circuit (IC) that deals with sensory or other analog input signals. Their performance and characteristics have a large repercussion on the signal processing carried out in the later (usual digital) stages of the IC, as distortions of the signal introduced in the ADC cannot usually be recovered. In general, ADCs because of their analog nature are handcrafted to achieve optimum characteristics for a given application. They usually require a wide range of custom analog circuit components, such as amplifiers, voltage/charge/current converters, integrators, addition/subtraction circuits, threshold switches, etc (van de Plassche, 2003).

However, this handcrafted, analog nature of ADCs is not in keeping with todays mostly digital Systems-on-Chip (SoC). SoCs due to their digital nature can be rapidly prototyped and transferred across technology nodes, something not possible with a handcrafted analog circuit. In addition, state-of-the-art deep-submicron technology nodes have become increasingly worse in their analog performance.

ADCs have started to partially follow this trend, offering architectures such as Delta-Sigma-Modulators (DSM) that only need low-performance analog components and move a large part of their functionality into the digital domain (Marijan and Ignjatovic, 2010; Mayr et al., 2010b). However, to really comply with the demands placed on modern ADCs, inspiration may be taken from a completely different domain, that of neural information processing and neuromorphic design. Neural networks rely for their overall function on multiple replication of a single, simple base element, the neuron. Thus, scaling and technology transfer of a neuromorphic ADC would be simplified. A neural network represents data across a population, thus inherently smoothing out variations and noise and making the signal representation more robust. Neurons take analog data as input, transferring it immediately into a pseudo-digital, timing based pulse representation. Thus, all subsequent processing would be digital directly after this first stage. Neural networks can replicate non-linear transfer functions of one or several input variables (Lovelace et al., 2010). Thus, sensor fusion and analog preprocessing could be achieved, which in conventional ADCs requires separate analog blocks (Chen et al., 2013).

This paper proposes using the Neural Engineering Framework (NEF) (Eliasmith and Anderson, 2004) as a method to build an ADC that incorporates most of the above advantages of neural networks. In the NEF, a signal is encoded across a neuron population by a set of encoder weights and the transfer functions of the neurons. A set of decoder weights can be computed that extracts the signal itself or a transformation of it from the postsynaptic current (PSC) traces of the neurons. By building the encoder step and the neurons in analog circuitry while having the decoding and signal reconstruction done in the digital domain, a straightforward conversion from analog to digital can be established.

Specifically, we show in this paper the usage of NEF as a linear, single input ADC comparable to conventional ADCs. The theoretical and simulative analysis is supported by an example design in a 180 nm CMOS technology, proving feasibility of the approach. The remainder of the paper is structured as follows:

section 2.1 introduces the NEF framework. In section 2.2, its general application to analog-digital-conversion is given. Section 2.4 details the analog and digital circuit design. Results are given in section 3.1 for an ADC based on idealized neurons in a neural network simulator. Results for the actual hardware implementation of neurons, encoder and decoder network are given in section 3.2. Section 4 discusses the significance of the results.

2. MATERIALS AND METHODS

2.1. REPRESENTATION OF ANALOG VARIABLES IN THE NEURAL ENGINEERING FRAMEWORK (NEF)

At its most basic, the NEF describes the transmission of an analog value or a set of values across a neuron population and its subsequent reconstruction from the neuron responses (see the upper part of **Figure 2**). In an abridged form, the theory is the following (Eliasmith and Anderson, 2004). A neuron population with a transfer function G is instantiated:

$$a_i = G\left(\alpha_i \cdot i_{syn,i} + b_i\right), \tag{1}$$

with $i_{syn,i}$ as input current, α_i as gain factor and b_i as offset. The transfer function can be e.g., that of a Leaky-Integrate-and-Fire neuron (LIAF), building a spike rate response a_i from $i_{syn,i}$. A vector variable $\vec{x}$ is then encoded in this synaptic current:

$$i_{syn,i} = \vec{e}_i \cdot \vec{x}. \tag{2}$$

The encoding vector $\vec{e}_i$ can be thought of as the preferred direction vector for that neuron: the vector for which that neuron will fire most strongly. To project the input vector into a sufficiently high-dimensional representation, α_i and b_i are varied between individual neurons. At the same time, allowing this variance in the neuron parameters enables a simple encoding vector composed of only discrete values. Usually, a binary vector consisting of +1 and -1 is chosen (Eliasmith and Anderson, 2004). Example tuning curves of neurons ($G\left(\alpha_i \cdot i_{syn,i} + b_i\right)$) are shown schematically in **Figure 1**.

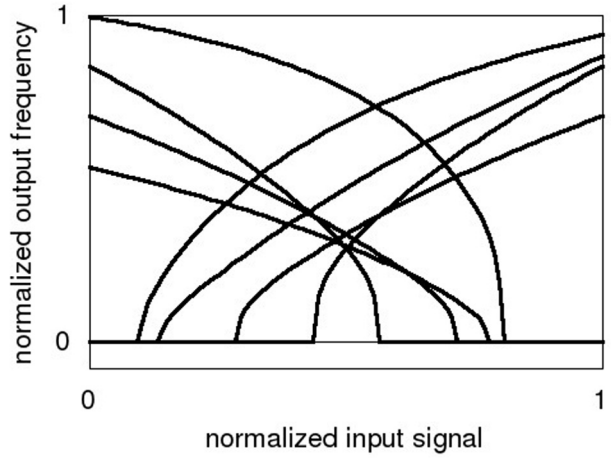


FIGURE 1 | Sample tuning curves for neurons in the NEF framework, normalized input signal $\bar{x}$ and normalized output frequency a_i.

While Equations 1 and 2 allow us to convert a vector $\vec{x}$ into neural activity a_i, it is also important to go the other way around. That is, given some neural activity, what value is represented? The simplest method is to find a linear decoder with decoder vector $\vec{d}_i$. This is a set of weights that maps the activity back into an estimate of $\vec{x}$, as follows:

$$\hat{\vec{x}} = \Sigma a_i \vec{d}_i. \tag{3}$$

For this, the neuron tuning curves are characterized across the input space $\vec{x}$. This is usually done in a regular raster. Specifically, for the scalar input x of the NEF ADC, 50 sample points spaced linearly across the normalized input range are applied as DC levels of 1 s duration and the neuron output rate measured. Given these characterized tuning curves, the optimal decoder weights for reconstructing $\vec{x}$ can be computed (Eliasmith and Anderson, 2004):

$$\vec{d} = \Gamma^{-1}\Upsilon \qquad \Gamma_{ij} = \Sigma_x a_i a_j \qquad \Upsilon_j = \Sigma_x a_j \vec{x} \tag{4}$$

The sum over x denotes the sum over the single characterization points of the tuning curves. The above matrix operation arrives at the least mean squared error fit for the decoder weights for a given transformation, as was also demonstrated in Mayr et al. (2008) for spectral reconstruction of a pixel sensor array (Henker et al., 2007). Please note that the decoder weight computation is given in Equation 4 for a linear reconstruction, but various non-linear transformations of $\vec{x}$ are also possible (Eliasmith and Anderson, 2004). The decoder weights $\vec{d}$ and an exponential postsynaptic current (PSC) kernel are then applied to each spike n of neuron i to arrive at the decoded signal $\hat{\vec{x}}$:

$$\hat{\vec{x}} = \sum_i \left\{ \sum_n \left[h_i(t - t_n) \right] \vec{d}_i \right\}$$
$$\text{with } h(t) = \frac{1}{\tau_{psc}} \cdot \Theta(t) \cdot e^{-\frac{t}{\tau_{psc}}}, \tag{5}$$

where $\Theta(t)$ is the step function. This theory can be extended to multiple networks and to symbol manipulation (Eliasmith and Anderson, 2004), but for our purposes, encoding a signal and decoding a transformation of that signal are sufficient.

2.2. AN ANALOG-DIGITAL-CONVERTER BASED ON THE NEF

The basic concept of using NEF as a single-channel ADC is shown in the lower part of **Figure 2**. The input vector $\vec{x}$ of Equation 2 is collapsed to a single scalar value $V_{in}(t)$. The initial step is to build a set of analog neurons that have varying tuning curves in both encoding directions (Equation 1). Then, these tuning curves are characterized and a set of decoding weights for a linear representation is computed (Equation 4). In operation, the analog input signal is applied to all neurons in parallel. The neurons feed their spikes into a synchronizer and a subsequent clocked digital decoder that operates on digitized versions of the decoder weights. An accumulator tree summarizes all spike contributions for a given clock cycle. Please note: Since the single PSC trains employed in Equation 5 are superimposed linearly and the exponential function is self-similar, the order of this computation

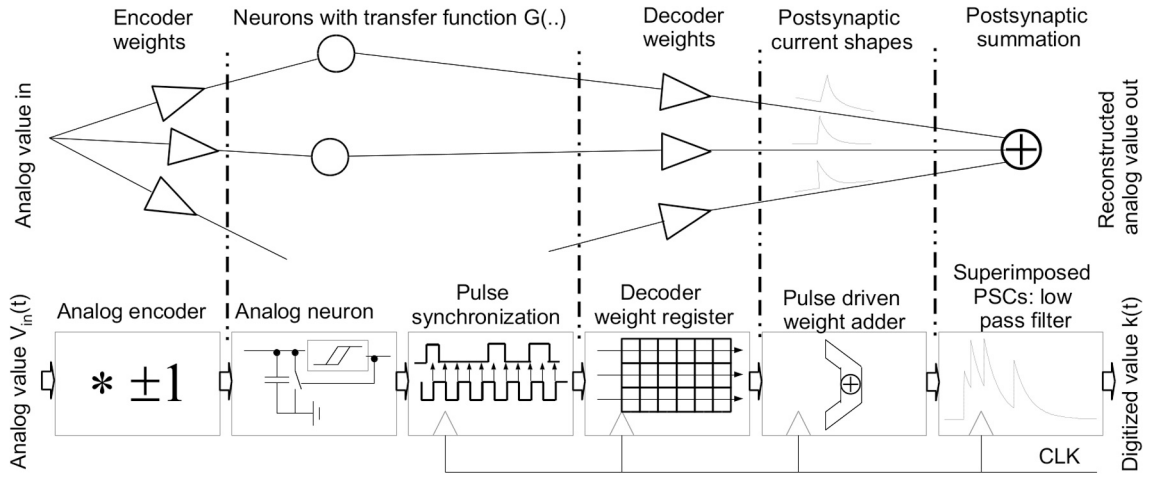


FIGURE 2 | Basic principle of NEF (upper diagram): encoding of an analog value, neuronal representation, decoding and signal reconstruction through an overlay of PSCs; Using NEF as an ADC (lower diagram): analog encoder and neuron representation, neuron pulse synchronization and digital decoder and lastly signal reconstruction via the PSC kernel.

can be commuted. Thus, in the NEF ADC, the exponential kernel is applied on the weighted spike summation as computed for each time step, thus simplifying the digital processing. This also eliminates the need for dedicated analog PSC circuits (Noack et al., 2011). The output k(t) of this exponentially decaying sum is the digital transformation of the analog value, i.e., the ADC output.

In essence, the transfer characteristic of the ADC is built up from the single neuron tuning curves via the decoder weights. Thus, the decoder parametrization gives the transfer characteristic of the ADC. Afterwards, a low-pass filter is applied through the PSCs to suppress the high-frequency components caused by the neuron pulses.

2.3. PERFORMANCE MEASURES FOR ADCs

To characterize the performance of the NEF ADC, comparison measures with conventional ADCs are required. The main characteristics of an ADC are its resolution (number of bits in each digital output word corresponding to an analog input sample), its sample rate (number of digital words representing analog values per second), its response to a DC step at the input and its conversion latency (i.e., time from analog input to digital conversion). As the NEF ADC does not follow a conventional ADC processing chain, these are not obvious in the current context. In section 3.1, we will derive analytical and empirical couterparts for these characteristics for the NEF ADC.

Besides these baseline characteristics, there are a number of performance figures that are usually employed to estimate the performance of an ADC. The effective number of bits (ENOB) is a measure where an analog DC signal is applied to the input and the sigma of the resulting histogram of output codes is computed (Baker, 2008). In essence, the ENOB computes the limit of the ADC resolution, that is the level where the output code moves from being correlated with the input signal toward noise. As the ENOB is a single DC level measure, it provides no information about the linearity of the transfer curve.

The transfer curve can be characterized by the integral nonlinearity (INL). For the INL, a ramp is applied to the input and the deviation of the overall transfer characteristic from the ideal one is computed (Provost and Sanchez-Sinencio, 2003). The maximum INL as a scalar measure provides information about the linearity limit of the ADC. As we will show later, a plot of the INL across the input DC level is also informative, as it shows the causes of the INL limits in terms of the NEF ADC design parameters.

The signal to noise-plus-distortion (SINAD) ratio uses a sine signal at the input, to measure the amplitude of the signal in the digital output minus the harmonics and noise (Baker, 2008). In this work, we have chosen INL and ENOB as the main performance indicators, as they capture a large part of the overall ADC characteristics and are easiest to simulate and extract.

2.4. OVERALL CIRCUIT DESIGN OF THE NEF ADC

The circuit design for the NEF ADC was carried out in a digital 180nm CMOS process, with a VDD of 1.8 V (digital and analog). The main goal of the circuit design is to transfer as much functionality into the digital domain as possible. Therefore, only the neurons are designed in analog circuitry, while decoder, adder and exponential decay are done digitally. The second goal is to incorporate a significant amount of runtime configurability. Therefore, the decoder weights, the PSC time constant and the sample rate are configurable. The size of the neuron population employed can also be adjusted by disabling some of the decoder weights.

2.5. THE NEURON CIRCUIT

The overall goal of the neuron circuit development is actually quite non-intutive: The transfer curves and therefore all analog parameters have to vary as much as possible to achieve a good coverage of the input dynamic range (Eliasmith and Anderson, 2004). The parameter variance introduced by manufacturing the IC in silicon, which is generally a detrimental effect, can be employed advantageously there. Since the NEF does not place specific demands on the qualitative neuron characteristic, basic

Integrate-and-Fire (IAF) neurons were chosen. The voltage input signal (see **Figure 2**) is applied to all neurons in parallel. The binary encoder of Equation 1 is realized by having two different types of neurons, one with a positive tuning curve and one with a negative curve (see **Figure 1**). **Figure 3** shows the schematic of the negative neuron, i.e., the neuron type where an increase in input voltage results in a lowering of the output frequency.

The neuron operates in asynchronous mode, with its pulse frequency determined solely by the input signal. At the left of the circuit, the signal enters at the gate of a source follower N1 and its current source N2. Both N1 and N2 are high threshold voltage types to reduce cross current. Through the V_{th} shift, the source follower extends the dynamic range of the subsequent voltage-current converter. The input range at N1 is rail to rail, i.e., GND to 1.8 V. Transistor P1 converts the offset input voltage to a current, with the source degeneration of P2 acting as a virtual increase of the gate length of P1. Again, P1 and P2 are high threshold voltage types. Thus, P1 and P2 can both be minimum sized for increased variation and still not draw excessive cross current. N4 through N6 and P3 (all minimum sized) generate a biasing current that is subtracted from itself via P4 and N7. This generates a bidirectional mismatch current which at N8 is added to the current caused by the input signal. The resulting current is mirrored via the (minimum sized) current mirror P5/P6 on the neuron membrane capacitance N9. N11, N12 and P7 form the inverter that defines the voltage threshold for neuron firing. As with P1, N12 is source degenerated to achieve a large virtual gate length which sets the threshold high and thus extends the dynamic range of the neuron membrane. Beneficially, this also decreases the cross current despite using minimum sized transistors. P8 and N13 delay and buffer the resulting pulse signal. A subsequent custom buffer stage further increases delay before the pulse resets the neuron membrane via N10.

A standard cell buffer shapes the pulse edges for output to the synchronuous digital part. In contrast to the variation-optimized analog part of the circuit, this pulse generation is designed in a more conventional way with minimized deviations. This stems from the fact that the pulse output signal has to conform to the timing specifications of the synchronous digital part (especially, a minimum low time between pulses and a minimum pulse length).

In terms of cross current and power, voltage biasing may have been the better choice for e.g., N2, P2 or N11. However, a design choice was to make the cell self-biased in preparation for inclusion in a digital cell library flow. Thus, the only analog input is the voltage to be digitized.

In terms of the tuning curve, mismatch at the current offset gives variation along the y-axis (compare **Figure 1**), while V_{th} deviation in the source follower results in x offset. The multiple current mirrors the signal is copied across and the gain deviation of P1 result in variation of the tuning curve slope.

Although all parts of the neuron operate above threshold of the CMOS transistors, subthreshold operation as in more conventional neuromorphic circuits (Bartolozzi and Indiveri, 2007) would actually be beneficial in this application, as variations are significantly higher than in above-threshold operation (Pineda de Gyvez and Tuinhout, 2004; Giulioni et al., 2012). However, only very slow ADCs would be possible this way (Yang and Sarpeshkar, 2006).

Figure 4 shows the layout of the neuron of **Figure 3**. The signal flow from left to right is similar to the schematic. No attention was given to matched layouting or similar, as variation should be maximized. The layout has been optimized for an eventual integration in a digital standard cell flow. Only one metal layer is used and the cell is compatible with the 5 μm Faraday digital standard cell raster.

2.6. THE DECODER AND PSC SUMMATION

The digital part of the NEF ADC is shown in **Figure 5**. It consists of synchronizer circuits for the input pulses of the analog neurons, decoder weight registers, an adder tree for summing all active decoder weights, and a low-pass filter.

The asynchronous pulses of the analog neurons have to be synchronized to the digital clock. Furthermore, the synchronized output should be independent of the neuron's pulse length. For this, a rising edge detector is used, which generates an output signal that is high for exactly one clock per pulse. For achieving full throughput, i.e., being able to detect one spike per clock cycle, the asynchronous pulse signal is sampled at both positive and negative clock edge via two standard two-stage synchronizers. An additional register stores the signal level at the previous

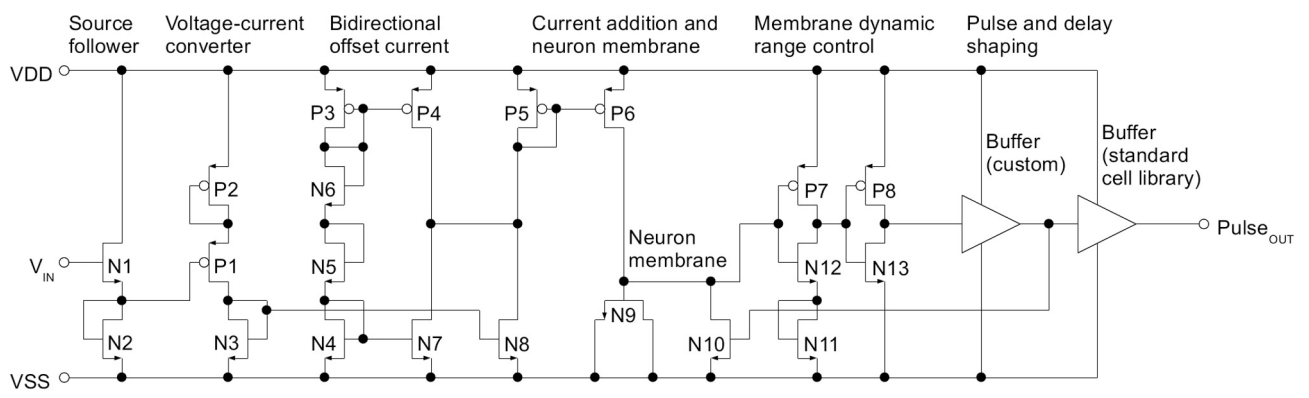


FIGURE 3 | Schematic of the analog neuron, type negative. For clarity, powerdown switching transistors have been omitted

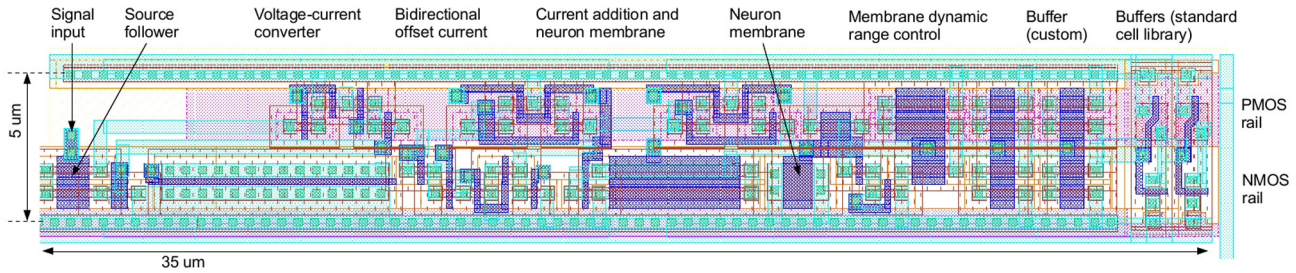


FIGURE 4 | Layout of the negative type neuron in UMC 180 nm CMOS technology, with signal flow from left to right similar to the schematic in Figure 3.

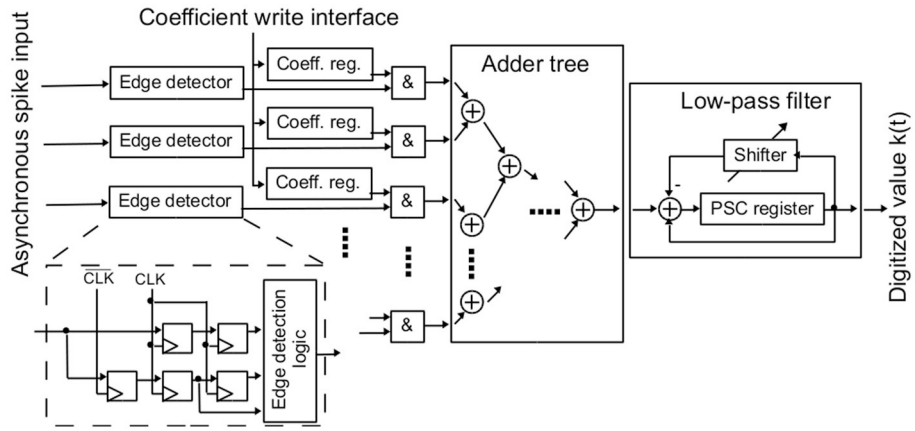


FIGURE 5 | Digital synchronous pulse registration, adder tree and low-pass filter.

negative clock edge. With this structure, a sequence negedge-posedge-negedge is available at the synchronizer output. An edge detection logic detects low-to-high changes in the input signal from this sequence.

The above edge detector captures all pulses in the asynchronous input signal, as long as the pulse length and the time the input is low between spikes is each higher than half a clock period. If the pulse length is shorter than this, only a fraction of the spikes is detected, attenuating the neuron's transfer function by a factor. In principle, the same happens if the input signal is low for a too short time between spikes. As this low time is decreasing with the pulse rate, the neuron's transfer function would start decreasing at high rates. While not intended, both effects would still be covered by the calibration of the NEF ADC.

As shown in **Figure 5**, each synchronized spike output activates its individual decoder weight register. The values of these registers are written via a separate configuration interface. The decoder weights effectively allow for setting up the transfer function of the AD conversion. The bit width of the decoder weights is a crucial parameter. The weight registers consume a significant part of the whole circuit area, so the bit width should be as small as possible. However, a certain minimum bit width is needed to not limit resolution of the AD conversion. In the current design, 8 bit signed values were used for achieving sufficient flexibility.

An adder tree calculates the sum over all active decoder weights. It was designed as a pipelined structure to achieve a throughput of one adder tree result per clock cycle. Computing across all spikes in a parallel manner as in the adder tree also obviates the need for any spike sorting or arbitration that would otherwise be required (Scholze et al., 2010)

The adder tree results are fed into the low-pass filter, resembling the PSC signal reconstruction. The low-pass filter result constitutes the output of the AD conversion. In each clock cycle, the current output of the adder tree is added to the low-pass filter's PSC register. At the same time, the current PSC register content is shifted right and subtracted, resulting in the desired first-order low-pass characteristic. The shift width b is configurable. The resulting PSC time constant τ_{psc} can be derived from the clock frequency f_{clk} and b by equating the result of the shift operation with an exponential decay:

$$\text{PSC(t)} \cdot \text{e}^{-\frac{1}{f_{\text{clk}} \cdot \tau_{psc}}} = \text{PSC(t)} \cdot (1 - 2^{-b}). \tag{6}$$

Applying the first order Taylor series approximation for small exponents to the left hand side of Equation 6, the following expression is derived

$$\tau_{psc} = \frac{2^b}{f_{\text{clk}}}. \tag{7}$$

As can be seen from Equation 7, realizing the PSCs in digital allows setting arbitrarily long time constants, which are necessary for a high-resolution ADC. Achieving the same in analog circuits would be difficult, especially in deep-submicron technologies (Noack et al., 2012).

The digital part of the NEF ADC was described in Verilog to be completely compliant with the standard digital design and synthesis flow. Thus, it can be easily ported between technologies.

3. RESULTS

The following two sections contain results of the NEF ADC based on neuron models simulated in Nengo and Spice simulations of the actual transistor-level neuron circuits. For quick reference, we give in **Table 1** the baseline ADC characteristics we use in both cases.

The baseline characteristics of the Nengo simulations are: $\tau_{PSC} = 128$ ms (i.e., a shift of 7 bit, compare Equation 7), decoder weight resolution $W_{res} = 8$ bit (compare section 2.6), maximum rate of the IAF neurons $f_{neuron,max} = 400$ Hz, and a population of $N_{neuron} = 512$. The spike times from the Nengo simulation are synchronized to the clock of the digital system model $f_{\mathrm{clk}} = 1$ kHz, i.e., the resolution of the pulse registration in the baseline is $T_{synch} = 1$ ms.

The baseline for the transistor-level simulations is the same as for the Nengo neurons, with the following modifications: The VDD of the neurons is 1.8 V, i.e., the normalized input signal $V_{in}(t)$ of **Figure 6** is mapped to a voltage swing of 0..1.8 V. The system model of the digital building blocks is sped up from the 1 kHz clock to 150 MHz, i.e., $T_{synch} = 6.67$ ns, to be compatible with the hardware neuron speed. The PSC time constant is adjusted by the same factor, i.e., a $\tau_{psc,biol} = 128$ ms in the Nengo simulations is equivalent to $\tau_{psc,tech} = 853$ ns in the transistor-level simulations. However, for comparison of the results of section 3.2 with section 3.1, the timebase is converted back to 1 kHz for all data plots except **Figure 10**.

A note on simulated time vs. execution time: The simulated time is the reference time of the simulation, which is biological real time in Nengo, i.e., if the PSC time constant is set to 30 ms, the PSC decays 63% in 30 ms simulated time. On the other hand, as CMOS circuit frequencies are inherently much higher, simulated time and all time constants can be chosen much shorter for these simulations. In actual hardware, this has the beneficial effect of increasing the conversion speed of the NEF ADC. On the other hand, execution time means the time it takes to run a certain input waveform on the network in either Nengo or Spice. The execution time is significantly less in Nengo as the simulator is optimized for neuron models and the neurons are abstracted to a set of equations. In contrast, the spice simulations deal with the transistor-level neurons and incorporate parasitic capacitances and resistors, which makes them significantly slower to execute.

A system model of the digital building blocks outlined in section 2.6 is used for the processing of the neuron output spikes of both the Nengo as well as the transistor level simulations. The system model has been verified against the synthesized Verilog code. The decoder weights are computed for a linear ADC characteristic for easy comparison with conventional ADCs.

Table 1 | Baseline characteristic of NEF ADC for Nengo and circuit level hardware simulations.

ADC	Baseline	
characteristic	**Nengo**	**Hardware**
τ_{PSC}	128 ms	853 ns
N_{neuron}	512	512
W_{res}	8 bit	8 bit
$f_{neuron,max}$	400 Hz	45 MHz
T_{synch}	1 ms	6.67 ns
Input range	idealized 0..1	GND to 1.8 V
Tuning curve spread	set by parameters	intrinsic through transistor mismatch

3.1. RESULTS OF AN IDEAL IMPLEMENTATION

To evaluate the efficacy of using the NEF framework as an ADC without carrying out analog hardware design, the initial implementation was done in the Nengo simulator (Stewart et al., 2009) with idealized neurons, having controlled tuning curve spread. IAF neurons are employed for compatibility to the hardware neurons, but there are negligible differences in results to e.g., LIAF neurons. Apart from allowing large parameter sweeps due to the reduced simulation time, using ideal simulated neurons also helps to establish a baseline performance that can be compared to the hardware neurons.

As can be seen in **Figure 6**, the waveform entered in the NEF ADC consist of an initial DC level for ENOB computation (0 to 4 s), DC level at 0 for ADC settling a the lower input limit (4 to 6 s) and a subsequent ramp for INL computation (6 to 10 s). The ramp is deliberately slow so that sections of it can be used as collection of quasi-DC levels at different input voltages to characterize the tuning curves of the neurons. The decoder weight vector is computed according to Equation 4 based on 50 input level sample points. Two important characteristics can already be

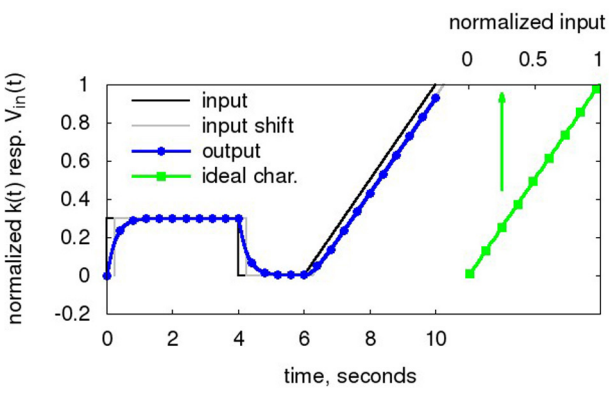


FIGURE 6 | Input waveform $V_{in}(t)$ (black). The Nengo simulator takes this normalized waveform as input. The digitized output k(t) (blue, circles), i.e., the state of the low-pass filter, is also normalized to 0..1. For comparison to the output, $V_{in}(t)$ shifted by τ_{PSC} is also displayed (black, dashed). Also shown is the ideal transfer characteristic (green, dashed, squares) as computed from the decoder weights and tuning curves in Equation 3. Baseline taken from **Table 1**, with τ_{PSC} of 256 ms for enhanced delay visibility.

observed: The digitized output has an exponential step response settling with τ_{PSC}. Also, the digitized output lags the input by τ_{PSC} at the input ramp, constituting the ADC latency.

Figure 7 shows a sample ENOB plot. The digitized output k(t) in the timespan from 2.9 to 3.4 s is subtracted from $V_{in}(t)$ and the difference plotted in a histogram. The ENOB is given by the standard deviation of this distribution (Baker, 2008). As can be seen, the NEF ADC output is similar to a conventional ADC, i.e., a DC level is replicated in the form of a narrow distribution of output codes around it. Despite the pulsing nature of the overall network and the high spread in decoder weight values (>20 max/min weight, i.e., a high amplification of some spike trains compared to others), there are no corresponding large transients in k(t).

Figure 8 shows a sample INL curve based on the ramp portion of the input waveform. For the low input voltages, the initial INL exhibits a residue from the settling to the ramp at $t = 6$ s. This is discounted for in the INL computation. The INL given in the following is the ±max deviation from the ideal curve (Provost and Sanchez-Sinencio, 2003), with respect to the normalized input range. The INL curve is not as characteristic as that of a more conventional ADC (Chae et al., 2013), as the transfer curve of the NEF ADC is built in a random fashion by the decoder weight computation based on the individual neuron deviations. The curve shown is representative for the NEF ADC, i.e., the INL curves are smooth but exhibit no characteristic shape. The ideal INL curve based on the transfer curve as computed from the decoder weights and tuning curves (compare **Figure 6**) is also shown. It can be observed that they match reasonably well, with the dynamic, ramp-based INL exhibiting additional high-frequency noise due to the network pulse activity. Increasing τ_{PSC} dampens the noise on the dynamic INL, reducing it to the level exhibited by the ideal INL. However, the ideal INL constitutes the lower bound, as it is determined largely by the number of neurons and thus is static and not amenable to further filtering.

The maximum INL for each datapoint shows a very steady 2 bit difference to the ENOB, e.g., the baseline example with ENOB 10.98 bit has a maximum INL of 8.91 bit. Unless otherwise noted, we will thus employ mainly ENOB as ADC performance characteristic, as it is more easily computed. **Table 2** details the behavior of the ENOB for a sweep of every variable given in the baseline description above. While some of these scaling characteristics of signal representation with network parameters have been explored for NEF (Choudhary et al., 2012), a full sweep of all relevant parameters has not been shown so far.

The scaling behavior of the ENOB can be extracted from three data points for each variable (baseline, example 1 and 2). Not surprisingly, there is linear scaling of ENOB with τ_{PSC}, $f_{neuron,max}$, and T_{synch}. All three variables affect the number of neuron pulses that are taken into account for a single output code to average over. This can be thought of as similar to scaling of resolution with the oversampling ratio (OSR) in a conventional first- order DSM (Perez et al., 2011). There is a saturation of ENOB with $f_{neuron,max}$ at about half the frequency given by T_{synch} (data not shown). This could be due to a saturation of the pulses per timestep, i.e., if there is on average more than one pulse per two timesteps, not much additional information is conveyed.

The ENOB scaling with N_{neuron} is worse than the scaling above for τ_{PSC}, as shown in **Figure 9**. So the naive assumption, that an increase in N_{neuron} results in a proportional increase of pulses for a given output code and thus gives linear scaling, does not hold. At the same time, it is slightly better than the expected $\sqrt{2}$ factor resulting from applying a signal to independent ADCs (King et al., 1998). The likely cause is that the neurons cannot be thought of as independent, as the ADC transfer characteristic is built from them and thus the decoder extracts the best fit transfer based on a combination of all of them.

The scaling of W_{res} also relates to the construction of the transfer characteristic: Surprisingly, there is almost no dependence between W_{res} and ENOB, i.e., the decoder weight can be quantized quite severly after computation and still result in a high-fidelity k(t). Intuitively, if the decoder weights have access to a widely varying neuron population, their own variation can

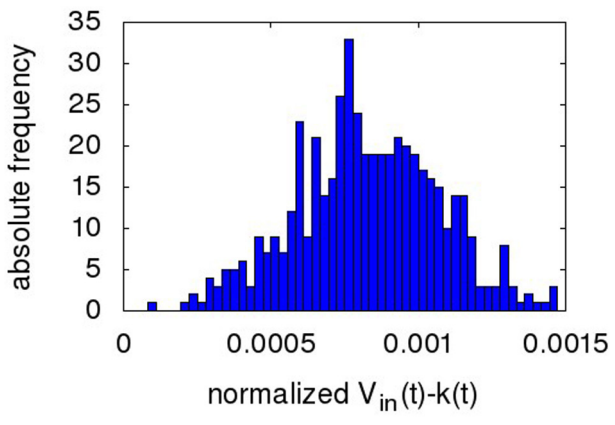


FIGURE 7 | Histogram of the digitized NEF output minus ideal analog input (both normalized to a dynamic range 0..1) during the time slice 2.9 to 3.4 s of the output wave in Figure 6 (i.e., the settled portion of the first DC input).

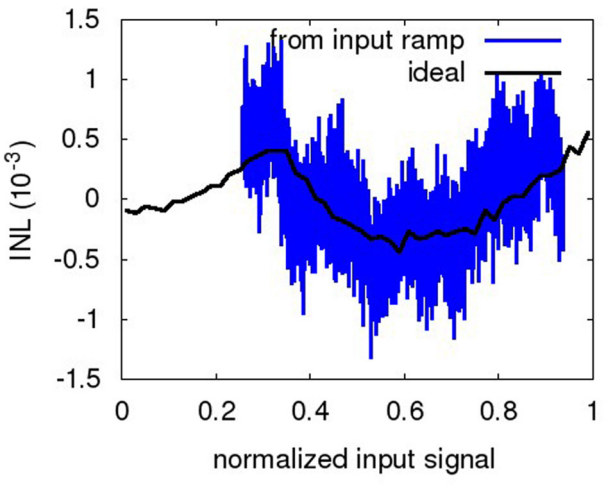


FIGURE 8 | INL of ramp portion of the waveform in Figure 6 (relative to the normalized full swing). The ideal INL based on the transfer curve of Equation 3 is also displayed (black, dashed).

Table 2 | Scaling of ENOB with design characteristic/variable.

ADC characteristic	Example 1	Example 2	ENOB scaling with characteristic	Remark
τ_{PSC}	8.98 bit$\|_{\tau_{PSC}=32\ ms}$	9.99 bit$\|_{\tau_{PSC}=64\ ms}$	linear	
N_{neuron}	8.16 bit$\|_{N_{neuron}=32}$	9.65 bit$\|_{N_{neuron}=128}$	*ca.* $\sqrt[1.5]{\ }$	
W_{res}	11.00 bit$\|_{W_{res}=5\ bit}$	10.92 bit$\|_{W_{res}=3\ bit}$	no dependence	See hardware discussion, has to be above a lower bound
$f_{neuron,max}$	7.69 bit$\|_{f_{neuron,max}=50\ Hz}$	9.73 bit$\|_{f_{neuron,max}=200\ Hz}$	linear	Scaling saturates at approx. $0.5 \cdot T_{synch}$
T_{synch}	6.81 bit$\|_{T_{synch}=2\ ms}$	5.90 bit$\|_{T_{synch}=4\ ms}$	linear	Please note: For this sweep, $f_{neuron,max} = 50$ Hz to avoid saturation

*Baseline characteristic as in **Table 1**, with resulting ENOB 10.98 bit.*

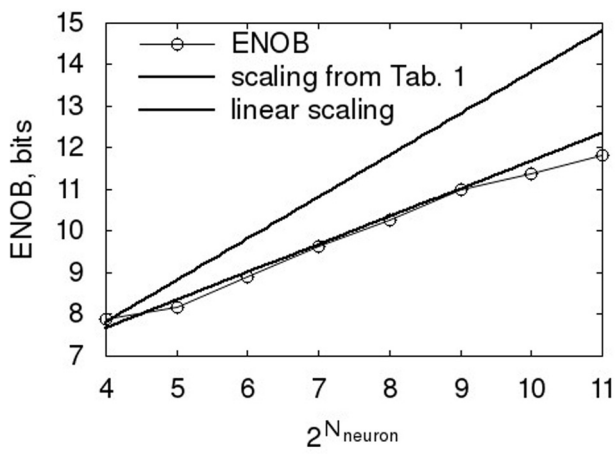


FIGURE 9 | Scaling of ENOB with the number of neurons. Baseline characteristic as in **Table 1**.

be very limited. There is only an empirical lower bound of W_{res} that has to be fulfilled to achieve reconstruction of $V_{in}(t)$ at all. This detail will be revisited in section 3.2.

The equivalent sample rate and Nyquist frequency are still missing from this characterization of the NEF ADC. The Nyquist signal frequency can be derived from the slew rate of a sine input signal, based on the assumption that it is reconstructed via τ_{PSC} with an exponentially decaying kernel. The sine has a maximum downward slew rate (at $t = 1/(2f)$), which can be equated to an exponentially decaying PSC starting at $t = 1/(2f)$ with an amplitude of 0.5:

$$\frac{\mathrm{d}\left\{0.5 \cdot \sin(2\pi f t)\right\}}{\mathrm{d}t}\Big|_{t=\frac{1}{2f}} = \frac{\mathrm{d}\left\{0.5 \cdot \mathrm{e}^{\frac{t-\frac{1}{2f}}{\tau_{PSC}}}\right\}}{\mathrm{d}t}\Big|_{t=\frac{1}{2f}}. \quad (8)$$

Solving this, we receive the maximum frequency $f_{sig,max}$ that a full-swing sine wave is supported by a given τ_{PSC}:

$$f_{sig,max} = \frac{1}{2\pi\,\tau_{PSC}}. \quad (9)$$

Table 3 | Characteristics of NEF used as a linear ADC.

Conversion rate	$\frac{1}{\tau_{PSC} \cdot \pi}$
Max. input frequency	$\frac{1}{2 \cdot \tau_{PSC} \cdot \pi}$
Empirical ENOB formula	$ENOB(Bit) = \mathrm{ld}(\tau_{PSC} \cdot \sqrt[1.5]{N_{neuron}} \cdot f_{neuron,max} \cdot \frac{1}{T_{reg,Pulse}} \cdot T_{norm})$
Conversion latency	τ_{PSC}
Settling time to a step response	$T_{set} = \tau_{PSC} \cdot ENOB \cdot \ln 2$
Dynamic range	rail-rail

With a corresponding Nyquist rate of $f_{sample} = 2 \cdot f_{sig,max}$, i.e., the frequency at which the state of the decaying accumulator is read out. Not entirely surprising, this constitutes a first order low-pass with cutoff at $1/\tau_{PSC}$.

The first-order low-pass characteristic also explains the conversion latency of the NEF ADC from a linear input ramp. The equations for the low-pass output y and input $x(t)$ are:

$$\tau_{PSC}\frac{\mathrm{d}y}{\mathrm{d}t} = -y + x(t) \quad \text{with} \quad x(t) = a \cdot t \quad (10)$$

The corresponding solution for the low-pass output is:

$$y(t) = a \cdot (t - \tau_{PSC}). \quad (11)$$

Thus, the low-pass output lags the input by τ_{PSC}. As can be seen from **Figure 6**, the other parts of the ADC processing chain do not add significantly to the conversion latency.

Table 3 sums up the results of this section, with an empirical ENOB formula based on **Table 2**. The ENOB formula is valid for $f_{synch} \geq 2 \cdot f_{neuron,max}$, i.e., for the neurons firing below saturation of the pulse registration. The scaling factor T_{norm} is approximately 0.6 ms. The dynamic range is a function of the neuron tuning curve variation, i.e., if the neurons have positive and negative responses that vary significantly even near the rails, the input can be rail-to-rail (compare **Figure 1** and **Figure 6**).

3.2. RESULTS OF THE CIRCUIT IMPLEMENTATION

This section expands the results obtained with the Nengo neurons to the neurons described in section 2.5. We use a neuron population that is based on Monte carlo variations of a parasitic

extraction of the layout in **Figure 4** and its counterpart for the positive neuron.

Figure 10 shows the time course of the decaying accumulator at its hardware timescale. From the zoom plot in the Figure, the single code values can clearly be seen. The small time constant configured in this example allows a clear view of the fine structure of the NEF ADC output. Due to the overlay of multiple single neuron transfer curves and the dynamic nature of the neurons, this output is quite stochastic, with the τ_{PSC} decay not readily evident. The code transitions cannot be identified, making more conventional INL measurement difficult (Baker, 2008).

In **Figure 11**, sample tuning curves of both types of neurons are overlaid for the Nengo generated neurons and the hardware neurons. When adjusting for the time base, the hardware neurons are somewhat slower than in Nengo, but the difference is not significant, as the ENOB starts to saturate at these frequencies in any case. It can be seen that in general, the circuit measures taken in section 2.5 for the hardware neurons generate a satisfactory range of offsets in x and y direction. The complete input range is converted with sufficiently varying neuron tuning curves, with the possible exception of a range close to the two rails, as the tuning curves there tend to correlate significantly and thus resolution would drop in these areas. The Monte Carlo models were set only to mismatch (i.e., not mismatch and process) to generate this curves, so this level of spread can be expected from a large part of manufactured IC instances. However, as the spread of the curves is determined by random effects of the manufacturing process, individual instances of the ADC have to be checked for sufficient spread, thus defining a yield in terms of ADC resolution. When comparing the two families of tuning curves, the main observation is that the Nengo generated neurons tend to vary more, especially in their gain.

As can be seen from the ENOB comparison in **Table 4**, this has a significant impact on the overall computation. If the neurons do not encode for sufficiently different features of the input signal, the representation of the input signal degrades. **Table 4** illustrates that the ENOB scaling with design characteristic is in general the same as in the Nengo simulations. However, the ENOB consistently is 1.6 bit less in the hardware. Consequently, the scaling factor T_{norm} in the empirical formula in **Table 6** is adjusted to approximately 0.2 ms.

The reduction in tuning curve variation also has an impact on the decoder weights. Due to the lower variation in tuning curves,

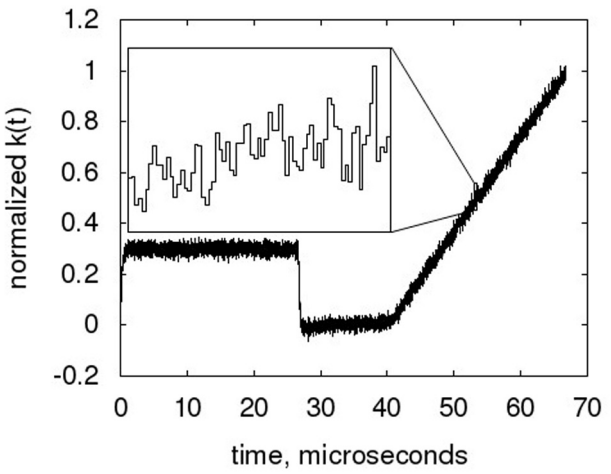


FIGURE 10 | Sample time course of the low-pass filter, with zoom of the ramp part of the waveform. The NEF ADC is configured to the hardware baseline characteristic, but with only 128 neurons and $\tau_{PSC} = 107$ ns (equiv. 16 ms) to reduce resolution and thus enhance the visibility of the curve progression from output code to output code.

Table 4 | ENOB-comparison for two examples Nengo and HW: τ_{PSC} sweep and N_{neuron} sweep.

NEF ADC parameters		Resulting ENOB	
τ_{PSC}	N_{neuron}	Nengo	Hardware
32 ms	128	7.64 bit	6.01 bit
64 ms	128	8.65 bit	6.99 bit
64 ms	512	9.99 bit	8.29 bit
128 ms	512	11.00 bit	9.29 bit

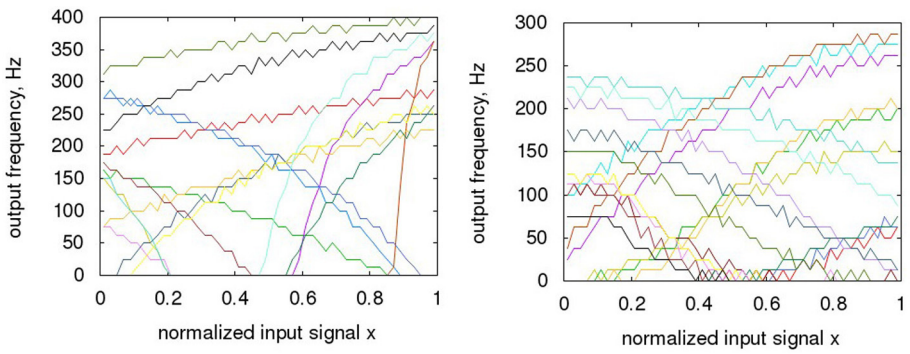


FIGURE 11 | Variation of neuron tuning curves; (A) Nengo; (B) hardware. For the hardware Monte Carlo variations, only mismatch is activated, as expected on a single die, i.e., no process deviations. Please note: the maximum rates of the hardware neurons are actually at ca. 45 MHz, which converts to 300 Hz when the timescale is converted to the Nengo one. The normalized input range of the hardware neurons corresponds to rail-rail, i.e., GND to 1.8 V.

the network needs a higher decoder weight precision in order to replicate a given transfer characteristic (as the transfer characteristic is built out of a combination of tuning curves and weight, see Equation 5). This can be seen from **Figure 12**, which shows the input waveform reproduction for Nengo and hardware neurons at 4 and 6 bit decoder weight resolution. At 4 bit, severe scaling errors exist for the hardware case in the wave output compared to the input.

Table 5 illustrates the quantitative INL and ENOB repercussions. The INL trend from the lower row of the plots in **Figure 12** is visible in the INL entries for 3 and 4 bit, where the hardware starts to worsen before the Nengo simulation. The INL as a measure based on a waveform better reflects this effect, the ENOB as a steady-state measure does not capture such dynamic errors sufficiently. Thus, there seems to exist a lower resolution limit for the decoder weights that is a function of the variation of the tuning curves. At 8 bit, the resolution chosen for the hardware implementation is well above this limit.

In order to test the robustness of the analog value representation to errors in the processing chain (neurons and decoder tree), we evaluate the failure or degradation of neurons:

- Random failure of one third of overall neurons, modeled through first computing a full decoder weight set, then setting one third of decoder weights to zero. For INL and ENOB, the output signal is adjusted for the corresponding amplitude decrease.
- Random perturbation of one third of positive neurons and one third of negative neurons, modeled by first computing a full decoder weight set, then randomly permuting the decoder weights of one third of the positive and negative neurons.

The ENOB shows interesting behavior: For the perturbation case, having the neuron in the network at all, even if with a different decoder weight, leaves the ENOB at its baseline level. That is, the spikes of this neuron still contribute to a less noisy DC level because they are added and low-pass filtered. However, the ENOB is degraded (with the amount expected from the formula in **Table 6**) if the neurons are lost, i.e., their spikes are not counted for the digitized output value.

As for the weight resolution, the INL is the more descriptive measure. **Figure 13** gives a representative example of the INL degradation due to neuron perturbation. The INL degradation for failure is similar both qualitatively and quantitatively. **Table 6** shows that the INL is degraded by about 1.2 bit for both neuron

Table 5 | Scaling of ENOB and INL with decoder weight resolution for Nengo simulation and hardware implementation.

	Decoder resolution		
	6 bit	**4 bit**	**3 bit**
ENOB Nengo	9.65 bit	9.60 bit	9.61 bit
INL Nengo	7.61 bit	7.63 bit	6.93 bit
ENOB Hardware	7.97 bit	7.95 bit	7.96 bit
INL Hardware	6.07 bit	5.69 bit	5.21 bit

Baseline hardware characteristic, but with 128 neurons.

Table 6 | Consequences of neuron failures in hardware: INL and ENOB for a random failure or perturbation of one third of the neurons.

	Baseline	**Neuron failure**	**Neuron perturbation**
ENOB	7.97 bit	7.64 bit	7.98 bit
INL	6.07 bit	4.83 bit	4.75 bit

Baseline hardware characteristic, neurons reduced to 128. ENOB and INL represent the average of 10 runs with different perturbations/failures.

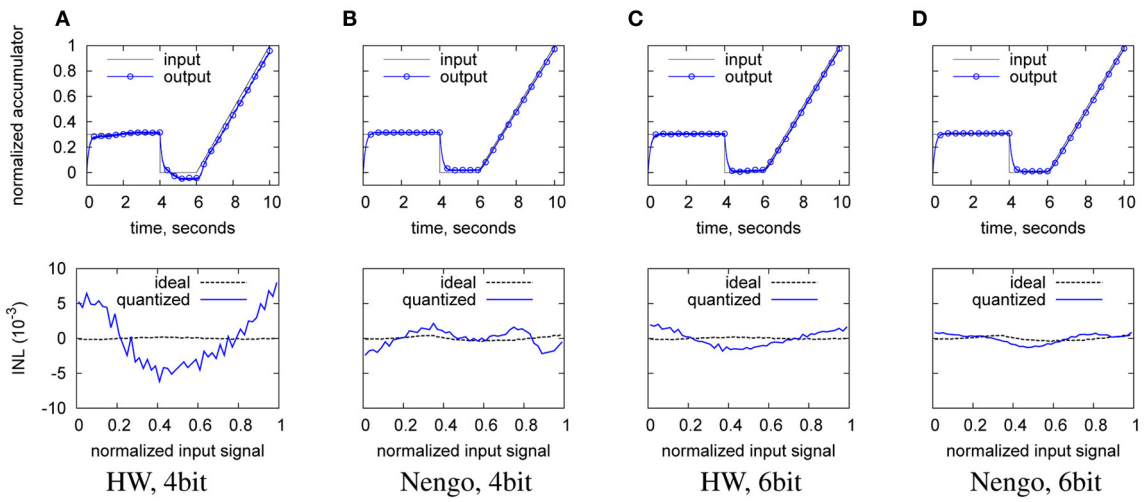


FIGURE 12 | Reproduction of the input waveform in the digitized output (top), and ideal INL (bottom), from left to right. (A) with hardware neurons and 4 bit decoder weight; **(B)** with Nengo neurons and 4 bit decoder weight; **(C)** with hardware neurons and 6 bit decoder weight; **(D)** with Nengo neurons and 6 bit decoder weight. Baseline hardware characteristic, but only 128 neurons.

failure and perturbation. Thus, not surprisingly, the transfer characteristic strongly depends on all neurons being present in the overall signal with the specific decoder weight that corresponds to their distinct manufacturing-given deviation in the tuning curve.

The INL degradation is actually more severe than evident from **Table 6**. As can be seen from **Figure 14**, the INL for the baseline is still dominated by pulse noise, while in **Figure 13**, static deviations clearly dominate. That is, for the baseline, a stronger low-pass-filtering could still decrease the INL, whereas for the perturbation case, further filtering does not diminish the INL.

However, judging from either **Table 6** or **Figure 13**, the INL degradation for this quite faulty network with one third disturbed neurons/weights is still only somewhere between 1.2 and 2.5 bit. Thus, this illustrates the soft degradation properties of the overall characteristic, which is due to the distributed analog value representation across the neurons. That is, there is not a single analog block that is crucial to the overall function, in contrast to e.g., the first amplifier in a pipeline converter. In case of neuron loss as simulated above, optimized rerouting could also be used to alleviate some of the loss (Mayr et al., 2007).

In **Tables 7–9**, the characteristics of the NEF ADC hardware design are summed up and compared to the state of the art. The final NEF ADC design contains 1280 neurons (640 of each encoding/type), operates at $f_{clk} = 150$ MHz and a VDD of 1.8 V. The area occupied by the digital building blocks is 2.69 mm^2, the area of the analog blocks (i.e., neurons) is 0.23 mm^2. Its analog power draw is 40 mW, digital 120 mW. As the digital blocks are designed to be runtime configurable, three different configurations are chosen for the comparison:

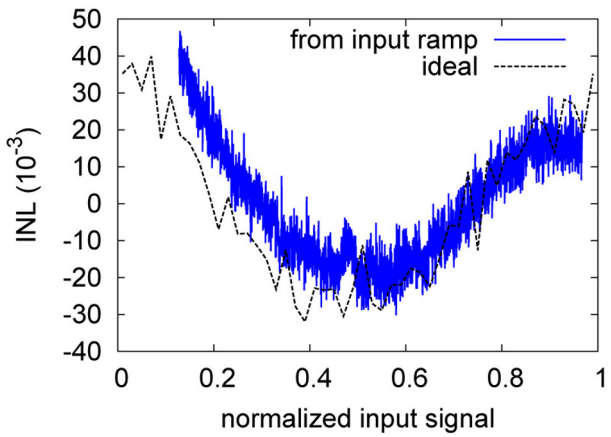


FIGURE 13 | Ideal and ramp-based INL of the neuron perturbation of Table 6.

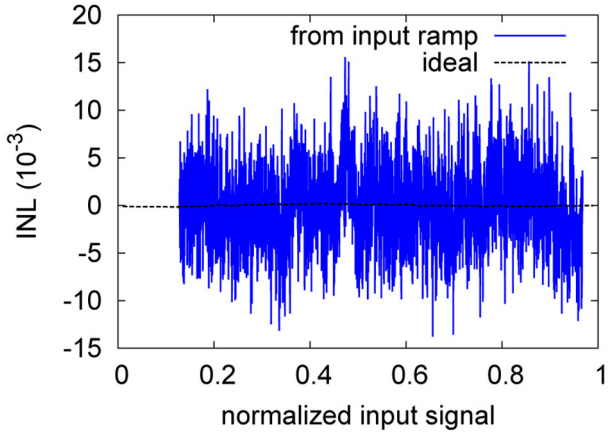


FIGURE 14 | Ideal and ramp-based INL of the baseline of Table 6.

Table 7 | High sample rate, low resolution comparison.

	This work	Weaver et al., 2011	Jain et al., 2012
Technology	180 nm	90 nm	130 nm
VDD	1.8 V	0.7 V	1.3 V
Power	160 mW	1.11 mW	4.0 mW
Area	2.92 mm^2	0.18 mm^2	0.38 mm^2
f_{sample}	12 MHz	21 MHz	15.625 MHz
ENOB/SNR	7.5 bit	5.8 bit	11.1 bit
FOM	74 pJ/ conv-step	0.95 pJ/ conv-step	0.11 pJ/ conv-step
Architecture	Neuromorphic parallel	Synthesized flash	high-speed DSM

Table 8 | Medium sample rate, medium resolution comparison.

	This work	Han et al., 2013	Perez et al., 2011
Technology	180 nm	180 nm	180 nm
VDD	1.8 V	0.45 V	1.5 V
Power	160 mW	1.35 μW	0.14 mW
Area	2.92 mm^2	–	0.48 mm^2
f_{sample}	750 kHz	200 kHz	200 kHz
ENOB/SFDR	11.5 bit	8.3 bit	13.6 bit
FOM	74 pJ/ conv-step	0.022 pJ/ conv-step	0.056 pJ/ conv-step
Architecture	Neuromorphic parallel	SAR	CT-DSM

Table 9 | Low sample rate, high resolution comparison.

	This work	Chae et al., 2013	Liu et al., 2013
Technology	180 nm	160 nm	mixed 500 and 180 nm
VDD	1.8 V	1.8 V	3.3 V
Power	160 mW	6.3 μW	0.28 mW
Area	2.92 mm^2	0.38 mm^2	1.14 mm^2
f_{sample}	730 Hz	25 Hz	10 kHz
ENOB/SFDR	21.5 bit	19.8 bit	17.4 bit
FOM	74 pJ/ conv-step	0.28 pJ/ conv-step	0.16 pJ/ conv-step
Architecture	Neuromorphic parallel	Zoom (SAR+DSM)	Incremental DSM

- Configuration for high sample rate: 12 MSamples/s, 7.5 bit ENOB (with a shift of 2 bit, i.e., equivalent $\tau_{psc,biol} = 4$ ms, actual $\tau_{psc,tech} = 26.7$ ns, compare Equation 7).
- Configuration for medium sample rate: 750 kSamples/s, 11.4 bit (shift of 6 bit, $\tau_{psc,biol} = 64$ ms,$\tau_{psc,tech} = 427$ ns).
- Configuration for low sample rate: 730 Samples/s, 21.4 bit (shift of 16 bit, $\tau_{psc,biol} = 65.5$ s,$\tau_{psc,tech}$=437 μs).

A common figure of merit (FOM) is used in the comparison that normalizes resolution, sampling rate and power (Walden, 1994). The state-of-the-art is chosen from the continuously updated survey in Murmann (2013). There is some debate whether power and area of the digital blocks of an ADC should be counted, as DSM comparisons usually leave out the decimation filter and other ADCs do not count their anti-aliasing filter (Murmann, 2013). However, our opinion is that since the digital components are an integral part of regular DSM ADCs and also of the presented NEF ADC, they should be included for a fair comparison, i.e., our comparison is based on 160 mW. This should be taken into account when viewing the FOM comparison with Jain et al. (2012) in **Table 7** and with Perez et al. (2011) in **Table 8**.

4. DISCUSSION

4.1. NEF IN A GENERAL NEUROMORPHIC VLSI CONTEXT

NEF has recently attracted significant interest from the neuromorphic community, with e.g., an implementation on Neurogrid (Choudhary et al., 2012). It exhibits several features of interest to engineers. Using it, one can engineer a neural system with a target reliable behavior based on unreliable elements. The target behavior can range from building blocks familiar to an engineer, such as control systems or filters (Dethier et al., 2013), up to abstract cognitive functions (Eliasmith, 2007). This paper has highlighted another useful aspect: NEF makes it easy to cross timing domains from asynchronous to synchronous and from analog to digital value representation. Traditionally, this has been one of the major bottlenecks when interfacing neuromorphic systems to more conventional processing units.

The other main challenge of neuromorphic engineering, i.e., achieving biological real time operation (Giulioni et al., 2012), could also be alleviated by NEF. By not representing the system variables directly as spikes, but rather abstracting the single pulses to a time-varying system state vector or scalar variable (Equation 5), the underlying neurons can be dictated by CMOS constraints (i.e., can be operated faster), while the state vector changes could be slower, i.e., able to interact with the outside world in biological real time. By adding this layer of abstraction on top of the neuromorphic network, the CMOS speed advantage can be utilized for e.g., a higher fidelty computation and/or representation of the system state variables, as shown in this paper. This layer of abstraction can also be used to transmit computational variables between neuromorphic units in a more CMOS-friendly fashion. Traditionally, states of neural networks are communicated by the single underlying spikes, requiring large bandwidths in FPGA-based spike routers (Hartmann et al., 2010) or even dedicated IC solutions (Scholze et al., 2011). By abstracting the single pulses to a time-varying digital state, bandwidth can be reduced significantly.

4.2. OTHER NEUROMORPHIC ADCs

There are a number of groups that have built ADCs based on neural networks. **Table 10** gives an overview of the salient features of these ADCs.

Some of those use time-invariant threshold neurons in architectures derived from conventional flash or pipeline ADCs (Chande and Poonacha, 1995). Neuromorphic principles have also been used to convert conventional architectures into the time domain. For example, Yang and Sarpeshkar (2006) show a pipeline ADC composed of Integrate-and-Fire (IAF) neurons that transfers the AD conversion into the time domain. While the use of subthreshold operation in Yang and Sarpeshkar (2006) makes for a very power efficient pipeline design, the entire design is targeted at a single application, without the wide configuration ability of the NEF ADC. For example, a higher resolution can only be achieved in the design of Yang and Sarpeshkar (2006) by increasing the complexity and power draw of the comparator. Also, a higher sample rate is only achievable through a non-subthreshold-operation of the neurons, loosing the energy advantage.

In both Chande and Poonacha (1995) and Yang and Sarpeshkar (2006) the performance of the design is ultimately limited by the precision of its handcrafted building blocks. Thus, no significant advantage is gained compared to conventional ADCs. In particular, both the above ADCs do not use the high parallelism of neural networks to increase robustness and/or conversion speed or precision. In contrast, another family of devices uses the noise shaping effect that a group of neurons achieves when recurrently inhibitory connected (Watson et al., 2004; Tapson and van Schaik, 2012). Here, the signal is represented robustly across a neuron population, i.e., the overall network activity is modulated by the signal (Mayr et al., 2009). The distribution across a neuron population even allows representation of signals above the intrinsic frequency of single neurons (Spiridon and Gerstner, 1999). One main drawback is that some of these architectures are unstable. There is also no fully established method to extract the digital output signal from such a network (Mayr and Schüffny, 2005).

4.3. NEF AS AN ANALOG-DIGITAL CONVERTER

The NEF ADC shares some characteristics with different conventional ADCs. For instance, time-domain ADCs also integrate the input signal to arrive at analog to time conversion that can then be digitized (Yang and Sarpeshkar, 2006). ADCs that oversample the input signal, such as the DSM mentioned in the introduction, also digitize an input signal with high frequency and low initial resolution. Similar to the NEF ADC, they build up resolution by removing high-frequency components with a filter. Also similar to a DSM, for most applications the NEF ADC does not require an analog Nyquist filter due to the low pass filter characteristic of the neurons and the PSC filter. The NEF ADC also shares some characteristics with flash ADCs, as both use a large parallelism of elements to arrive at a coarse fast quantization. Similar to the NEF ADC, some flash ADCs also rely on statistical deviation of elements for their quatization curve Weaver et al., 2011).

The comparison across **Tables 7–9** shows that in terms of absolute figures of sample rate and bit resolution achieved,

the NEF ADC is competitive. However, it underperforms quite severly with regard to area and power, see the FOM comparison.

The major part of the area of the NEF ADC is spent on the digital building blocks, letting it benefit significantly from technology scaling. Conventional ADCs do not shrink well due to their usually significantly larger portion of analog circuitry. Thus, the area comparison would look decidedly different in e.g., a 28 nm technology, where the digital blocks would only occupy approx. 0.080 mm^2. Also, a large fraction of the digital area is spent on the conservative choice of the decoder weight resolution, the large width of the decaying accumulator and the reconfiguration options. Thus, a more dedicated, less configurable design would realize additional area savings. The analog neurons can also be shrunk with the technology node, as this increases their speed and amplifies their mismatch, both desirable properties for the NEF ADC.

Pushing the power consumption of the NEF ADC into a competitive range is harder than for the area. However, as the design of the NEF ADC is intended as a proof-of-principle, no effort has been spent on power optimization. Especially the neuron power draw is quite excessive, with its multiple current paths from VDD to ground. More that 80% of its power draw is not spent on charging the membrane or for switching, but in the offset and gain error stages. Due to downscaling, future neurons in smaller technologies may offer the same variation with significantly less involved circuits, i.e., less power budget. The digital circuitry has also not been optimized for low power draw. Since the NEF is robust to small timing variations in its pulses, the initial digital building blocks such as the decoder weight readout and adder tree could be run asynchronously, only synchronizing directly before the decay register. This would save significant power in the clock tree. For overall clocking, energy-efficient variable clock generators (Eisenreich et al., 2009) could be used to adjust the operating frequency of the system, making a system possible that offers the same resolution at different sample rates, similar to (Yip and Chandrakasan, 2011). Also, the multiple configuration options and corresponding bit widths at all stages add to the power draw. Here, gating techniques that shut off parts of the circuitry not needed for a given configuration have to be explored.

In terms of absolute performance figures, **Table 9** shows that the NEF ADC may be especially competitive when it comes to achieving very high resolution digitization, as resolution can be achieved cheaply by digitally averaging over a longer time span. This aspect will be preferentially evaluated once the hardware is available.

However, while a one-to-one comparison of the NEF ADC with conventional ADC is informative, it was not the single design target. The main advantages of the NEF used as an ADC are the following:

- In the NEF ADC, the signal is represented in a robust way across a neuron population (see **Table 6**). Since the network is purely feed-forward, stability is not an issue.
- NEF makes little demand on the specific transfer characteristics of the analog neurons, and the encoder network

Table 10 | Comparison of various neuromorphic ADC concepts.

References	Description	Most similar conventional architecture	Parallel/ serial	Required analog precision / required design effort	Config.	Power	Sensor fusion possible
This work	digital decoding of analog input from neuron population signals	Flash and feedforward oversampling	Parallel	Very low / low, repetitive neuron circuit	Rate and resolution	x10 more power than best reported conventional	Yes
Tapson and van Schaik, 2012	Parallel noise shaping network with lateral inhibition	DSM	Parallel	Low / low, repetitive neuron circuit	rate and resolution	No data	No data
Chande and Poonacha, 1995	Binary threshold neurons in a weighted MSB to LSB decoder network	Successive approximation	Serial	Equal to ADC resolution / low, repetitive neuron circuit	resolution	No data, but likely comparable to median conventional	Yes
Yang and Sarpeshkar, 2006	Time-domain pipeline architecture, with neurons handling time domain processing	Pipeline	Serial	Equal to ADC resolution / high, numerous handcrafted components	No	on par with best reported conventional, subthreshold operation	No

uses binary weights. Accordingly, no high-fidelity, complex analog circuits are required anywhere in the system. The handcrafted analog circuits usually needed for an ADC are reduced to two simple neuron circuits, that are multiply instantiated.
- A large part of the processing is carried out in digital, making technology scaling very attractive and enabling design transfer across technologies with minimum effort.
- The possibility of adjusting the transfer characteristic, resolution and sample rate at runtime make for a very flexible system. In addition, the NEF framework incorporates a simple method to input several signals into this network and do computation with them for e.g., sensor fusion.

In addition, NEF represents a theoretically well-explored paradigm, coming complete with a mathematically rigorous method for high-fidelity extraction of the original signal (Eliasmith and Anderson, 2004). Scaling and signal representation behavior necessary to achieve a given target ADC characteristic has been partially established in Choudhary et al. (2012) and treated in depth in this manuscript.

4.4. LIMITS OF THE NEF ADC RESOLUTION

The INL plots of section 3.2 illustrate how insufficient decoder weight resolution, insufficient neuron number or tuning curve variation (represented by setting decoder weights zero) or insufficient tuning curve characterization (represented by perturbed decoder weights) can negatively influence the static INL. Especially for the case of perturbed decoder weights, the ENOB does not provide sufficient characterization of the ADC characteristic, as it stays virtually constant. The INL plots on the other hand provide a clear indication that static INL dominates dynamic INL (i.e., the INL caused by incomplete filtering as seen in the waveform-based INL in **Figure 8**). As can be seen from **Table 2**, increasing the number of neurons increases resolution only sub-linear, while power draw increases linearily. Thus, an ideal NEF ADC should be operated at the border between the dynamic INL and the static INL (also **Figure 8**). In other words, tuning curve variation, decoder weight resolution and especially neuron number should just be sufficient for the target INL, with τ_{PSC} chosen such that the remaining pulse noise is on the same order as the static INL.

The above is valid if the NEF ADC is built for a single conversion characteristic. In contrast, when using the NEF ADC over a wide range of possible τ_{PSC}, there are two different options. Either the number of neurons is chosen very large so that even for the high resolution at large τ_{PSC}, a sufficiently linear overall transfer characteristic can be constructed from the neuron tuning curves. However, this implies that at small τ_{PSC}, the number of neurons is far in excess of those needed and the NEF ADC is dominated by pulse noise. The second option would be to choose the number of neurons only sufficient for linearity at small τ_{PSC}, i.e., at low resolutions. At high resolutions (large τ_{PSC}), the static INL would intentionally dominate. To still achieve linearity, the digital output codes of the low pass filter would be passed through a look-up-table containing the inverse of the static INL curve.

4.5. OUTLOOK

In the current version, the NEF ADC still has a number of drawbacks. It is very susceptible to temperature and VDD variation. The transfer characteristic must thus ideally be measured for all these operating conditions and stored, or a constant on-line characterization has to be carried out. Built-in self-tests (BIST) such as Flores et al. (2004) look promising, as they would allow enhancing the NEF ADC with a constant self-monitoring at very little reduction of usable sample rate. Especially digital-heavy versions of BIST could be incorporated with little detriment in design time, as most of the functionality would be synthesizable. The area overhead would also be minimal if the NEF ADC is used as part of a larger digital system where existing compute resources could be reused for BIST (Flores et al., 2004).

A second, more experimental approach might be to adjust the decoder weights online via neuromorphic means, such as synaptic plasticity. NEF has been shown to be amenable to supervised biologically plausible plasticity rules which have as supervisory input the overall transfer characteristic (Bekolay et al., 2013). This plasticity could act either in the analog domain as adjustable factor in the single neuron processing chains, or it could act directly on the digital decoder weights. A candidate plasticity rule that can be configured for a wide range of behavior, i.e., for different compensation or decoder characteristics, has recently been demonstrated (Mayr and Partzsch, 2010) and implemented efficiently in analog CMOS hardware (Mayr et al., 2010a). Digital plasticity rules have been shown e.g., on the Spinnaker system (Jin et al., 2010).

The main point for future work, however, will be to take advantage of the computational capability inherent in NEF. In this paper, NEF has been reduced to a linear representation of a single variable. We will explore various non-linear ADC characteristics and joint conversion of multiple inputs, offering complex sensor fusion and feature extraction (König et al., 2002; Mayr and Schüffny, 2007). Beyond the usage as ADC, the NEF could pave the way toward a future mixed-signal, mixed neuromorphic/conventional system on chip. The NEF could take various elements (regular CMOS, memristors (Jo et al., 2010; Ou et al., 2013), other nanoscale elements) and engineer a system with a set of target computations based on these elements. As demonstrated, such a framework can easily cross the barrier between asynchronous and synchronous systems as well as between analog and digital domains, doing the signal reconstruction either digitally as demonstrated here or via compact, configurable analog PSC circuits (Noack et al., 2010). Signal reconstruction could be via a decoder learned in memristors (Mayr et al., 2012). Thus, one could employ each type of system/device where it is most beneficial and arrive at an amalgan of the state of the art in the neuromorphic discipline, the digital/analog CMOS discipline and in nanodevice systems.

ACKNOWLEDGMENTS

We would thank T. Stewart, C. Eliasmith and F. Corradi for fruitful discussions on the Neural Engineering Framework. G. Indiveri has been helpful in pointing out examples of neuromorphic ADCs. This research has received funding from the European Union Seventh Framework Programme (FP7/2007- 2013) under grant agreement no. 269459 (CORONET).

REFERENCES

Baker, B. (2008). *A Glossary of Analog-to-Digital Specifications and Performance Characteristics.* Technical Report SBAA147A, Texas Instruments.

Bartolozzi, C., and Indiveri, G. (2007). Synaptic dynamics in analog VLSI. *Neural Comput.* 19, 2581–2603. doi: 10.1162/neco.2007.19.10.2581

Bekolay, T., Kolbeck, C., and Eliasmith, C. (2013). "Simultaneous unsupervised and supervised learning of cognitive functions in biologically plausible spiking neural networks," in *Proceedings of the 35th Annual Conference of the Cognitive Science Society* (Berlin), 169–174. Available online at: http://mindmodeling.org/cogsci2013/papers/0058/paper0058.pdf

Chae, Y., Souri, K., and Makinwa, K. (2013). "A 6.3 uW 20b incremental zoom-adc with 6ppm INL and 1 uV offset," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International* (San Francisco, CA), 276–277.

Chande, V., and Poonacha, P. G. (1995). On neural networks for analog to digital conversion. *IEEE Trans. Neural Netw.* 6, 1269–1274. doi: 10.1109/72.410371

Chen, W.-M., Chiueh, H., Chen, T.-J., Ho, C.-L., Jeng, C., Chang, S.-T., et al. (2013). "A fully integrated 8-channel closed-loop neural-prosthetic SoC for real-time epileptic seizure control," in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (San Francisco, CA), 286–287. doi: 10.1109/ISSCC.2013.6487737

Choudhary, S., Sloan, S., Fok, S., Neckar, A., Trautmann, E., Gao, P., et al. (2012). Silicon neurons that compute. in *Artificial Neural Networks and Machine Learning - ICANN 2012*, Vol. 7552 of *Lecture Notes in Computer Science*, (Springer Berlin Heidelberg), 121–128.

Dethier, J., Nuyujukian, P., Ryu, S. I., Shenoy, K. V., and Boahen, K. (2013). Design and validation of a real-time spiking-neural-network decoder for brain–machine interfaces. *J. Neural Eng.* 10:036008. doi: 10.1088/1741-2560/10/3/036008

Eisenreich, H., Mayr, C., Henker, S., Wickert, M., and Schüffny, R. (2009). A novel ADPLL design using successive approximation frequency control. *Elsevier Microelectron. J.* 40, 1613–1622. doi: 10.1016/j.mejo.2008.12.005

Eliasmith, C. (2007). How to build a brain: from function to implementation. *Synthese* 159, 373–388. doi: 10.1007/s11229-007-9235-0

Eliasmith, C., and Anderson, C. C. H. (2004). *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. Cambridge, MA: MIT Press.

Flores, M., Negreiros, M., Carro, L., and Susin, A. (2004). INL and DNL estimation based on noise for ADC test. *Instrum. Meas. IEEE Trans.* 53, 1391–1395. doi: 10.1109/TIM.2004.834096

Giulioni, M., Camilleri, P., Mattia, M., Dante, V., Braun, J., and Del Giudice, P. (2012). Robust working memory in an asynchronously spiking neural network realized in neuromorphic VLSI. *Front. Neurosci.* 5:149. doi: 10.3389/fnins.2011.00149

Han, D., Zheng, Y., Rajkumar, R., Dawe, G., and Je, M. (2013). "A 0.45V 100-channel neural-recording IC with sub- uW/channel consumption in 0.18 um CMOS," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International* (San Francisco, CA), 290–291.

Hartmann, S., Schiefer, S., Scholze, S., Partzsch, J., Mayr, C., Henker, S., et al. (2010). "Highly integrated packet-based AER communication infrastructure with 3Gevent/s throughput," in *Proceedings of IEEE International Conference on Electronics, Circuits, and Systems ICECS10* (Athens), 952–955.

Henker, S., Mayr, C., Schlüssler, J.-U., Schüffny, R., Ramacher, U., and Heittmann, A. (2007). "Active pixel sensor arrays in 90/65nm CMOS-technologies with vertically stacked photodiodes," in *Proceedings of IEEE International Image Sensor Workshop IIS07* (Maine), 16–19.

Jain, A., Venkatesan, M., and Pavan, S. (2012). Analysis and design of a high speed continuous-time delta sigma modulator using the assisted opamp technique. *Solid State Circ. IEEE J.* 47, 1615–1625. doi: 10.1109/JSSC.2012.2191210

Jin, X., Rast, A., Galluppi, F., Davies, S., and Furber, S. (2010). "Implementing spike-timing-dependent plasticity on spinnaker neuromorphic hardware," in *Neural Networks (IJCNN), The 2010 International Joint Conference on IEEE* (Barcelona), 1–8.

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301. doi: 10.1021/nl904092h

King, E., Eshraghi, A., Galton, I., and Fiez, T. (1998). A nyquist-rate delta-sigma A/D converter. *Solid State Circ. IEEE J.* 33, 45–52. doi: 10.1109/4.654936

König, A., Mayr, C., Bormann, T., and Klug, C. (2002). "Dedicated implementation of embedded vision systems employing low-power massively parallel feature computation," in *Proceeding of the 3rd VIVA-Workshop on Low-Power Information Processing* (Chemnitz), 1–8.

Liu, Y., Bonizzoni, E., D'Amato, A., and Maloberti, F. (2013). "A 105-dB SNDR, 10 kSps multi-level second-order incremental converter with smart-DEM consuming 280 uW and 3.3-V supply," in *ESSCIRC (ESSCIRC), 2013 Proceedings* (Bucharest), 371–374. doi: 10.1109/ESSCIRC.2013.6649150

Lovelace, J., Rickard, J., and Cios, K. (2010). "A spiking neural network alternative for the analog to digital converter," in *Neural Networks (IJCNN), The 2010 International Joint Conference On* (Barcelona), 1–8. doi: 10.1109/IJCNN.2010.5596909

Marijan, M., and Ignjatovic, Z. (2010). "Code division parallel delta-sigma AD converter with probabilistic iterative decoding," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)* (Paris), 4025–4028. doi: 10.1109/ISCAS.2010.5537642

Mayr, C., Ehrlich, M., Henker, S., Wendt, K., and Schüffny, R. (2007). Mapping complex, large-scale spiking networks on neural VLSI. *Int. J. Appl. Sci. Eng. Technol.* 4, 37–42.

Mayr, C., Henker, S., Krause, A., Schlüssler, J.-U., and Schüffny, R. (2008). "65 nm CMOS sensors applied to mathematically exact colorimetric reconstruction," in *Proceedings IASTED International Conference on Computer Graphics and Imaging CGIM 08*, 56–63.

Mayr, C., Noack, M., Partzsch, J., and Schüffny, R. (2010a). "Replicating experimental spike and rate based neural learning in CMOS," in *IEEE International Symposium on Circuits and Systems ISCAS 2010*, 105–108.

Mayr, C., and Partzsch, J. (2010). Rate and pulse based plasticity governed by local synaptic state variables. *Front. Synaptic. Neurosci.* 2:33. doi: 10.3389/fnsyn.2010.00033

Mayr, C., Partzsch, J., and Schüffny, R. (2009). Transient responses of activity-dependent synapses to modulated pulse trains. *Elsevier Neurocomput.* 73, 99–105. doi: 10.1016/j.neucom.2009.02.019

Mayr, C., Scholze, S., Ander, M., Henker, S., and Schüffny, R. (2010b). "Aliasing-free variable gain delta sigma modulator for use in an analog frontend," in *17th International Conference on Mixed Design of Integrated Circuits and Systems MIXDES 2010* (Wrocław), 195–199.

Mayr, C., and Schüffny, R. (2005). Applying spiking neural nets to noise shaping. *IEICE Trans. Inf. Syst.* E88-D, 1885–1892. doi: 10.1093/ietisy/e88-d.8.1885

Mayr, C., and Schüffny, R. (2007). "Neighborhood rank order coding for robust texture analysis and feature extraction," in *IEEE Computer Society, Proceedings of the 7th International Conference on Hybrid Intelligent Systems HIS 07* (Kaiserslautern), 290–301. doi: 10.1109/HIS.2007.12

Mayr, C., Stärke, P., Partzsch, J., Cederstroem, L., Schüffny, R., Shuai, Y., et al. (2012). Waveform driven plasticity in BiFeO3 memristive devices: Model and implementation. *Adv. Neural Inf. Proc. Syst.* 25, 1700–1708. Available online at: http://papers.nips.cc/paper/4595-waveform-driven-plasticity-in-bifeo3-memristive-devices-model-and-implementation.pdf

Murmann, B. (2013). "ADC performance survey 1997–2013," in *Technical Report* (Stanford, CA: Stanford University).

Noack, M., Mayr, C., Partzsch, J., and Schüffny, R. (2011). "Synapse dynamics in CMOS derived from a model of neurotransmitter release," in *20th European Conference on Circuit Theory and Design ECCTD2011* (Linkoping), 198–201. doi: 10.1109/ECCTD.2011.6043316

Noack, M., Mayr, C., Partzsch, J., Schultz, M., and Schüffny, R. (2012). "A switched-capacitor implementation of short-term synaptic dynamics," in *19th International Conference on Mixed Design of Integrated Circuits and Systems MIXDES 2012* (Warsaw), 214–218.

Noack, M., Partzsch, J., Mayr, C., and Schüffny, R. (2010). "Biology-derived synaptic dynamics and optimized system architecture for neuromorphic hardware," in *17th International Conference on Mixed Design of Integrated Circuits and Systems MIXDES 2010*, 219–224.

Ou, X., Luo, W., Du, N., Wu, C., Zhang, W., Burger, D., et al. (2013). Nonvolatile multilevel resistive switching in Ar+ irradiated BiFeO3 thin films. *IEEE Electr. Device Lett.* 34, 54–56. doi: 10.1109/LED.2012.2227666

Perez, A., Bonizzoni, E., and Maloberti, F. (2011). "A 84dB SNDR 100kHz bandwidth low-power single op-amp third-order Delta Sigma modulator consuming 140 uW," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International* (San Francisco, CA), 478–480. doi: 10.1109/ISSCC.2011.5746405

Pineda de Gyvez, J., and Tuinhout, H. (2004). Threshold voltage mismatch and intra-die leakage current in digital CMOS circuits. *Solid State Circ. IEEE J.* 39, 157–168. doi: 10.1109/JSSC.2003.820873

Provost, B., and Sanchez-Sinencio, E. (2003). On-chip ramp generators for mixed-signal BIST and ADC self-test. *IEEE J. Solid State Circ.* 38, 263–273. doi: 10.1109/JSSC.2002.807415

Scholze, S., Eisenreich, H., Höppner, S., Ellguth, G., Henker, S., Ander, M., et al. (2011). A 32 GBit/s communication SoC for a waferscale neuromorphic system. *Integr. VLSI J.* 45, 61–75. doi: 10.1016/j.vlsi.2011.05.003

Scholze, S., Henker, S., Partzsch, J., Mayr, C., and Schüffny, R. (2010). "Optimized queue based communication in VLSI using a weakly ordered binary heap," in *17th International Conference on Mixed Design of Integrated Circuits and Systems MIXDES 2010* (Wrocław), 316–320.

Spiridon, M., and Gerstner, W. (1999). Noise spectrum and signal transmission through a population of spiking neurons. *Network* 10, 257–272. doi: 10.1088/0954-898X/10/3/304

Stewart, T., Tripp, B., and Eliasmith, C. (2009). Python scripting in the Nengo simulator. *Front. Neuroinform.* 3:9. doi: 10.3389/neuro.11.007.2009

Tapson, J., and van Schaik, A. (2012). "An asynchronous parallel neuromorphic ADC architecture," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on* (Seoul), 2409–2412. doi: 10.1109/ISCAS.2012.6271783

van de Plassche, R. (2003). *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*. Dordrecht: Kluwer Academic Publishers.

Walden, R. H. (1994). "Analog-to-digital converter technology comparison," in *Proceeding of the GaAs IC Symposium* (Philadelphia, PA), 228–231.

Watson, B. C., Shoop, B. L., Ressler, E. K., and Das, P. K. (2004). Analog-to-digital conversion using single-layer integrate-and-fire networks with inhibitory connections. *EURASIP J. Adv. Signal Proces.* 2004:894284. doi: 10.1155/S1110865704405083

Weaver, S., Hershberg, B., and Moon, U.-K. (2011). "Digitally synthesized stochastic flash ADC using only standard digital cells," in *2011 Symposium on VLSI Circuits (VLSIC)* (Honolulu, HI), 266–267.

Yang, H., and Sarpeshkar, R. (2006). A bio-inspired ultra-energy-efficient analog-to-digital converter for biomedical applications. *IEEE Trans. Circ. Syst I Reg. Pap.* 53, 2349–2356. doi: 10.1109/TCSI.2006.884463

Yip, M., and Chandrakasan, A. (2011). "A resolution-reconfigurable 5-to-10b 0.4-to-1V power scalable SAR ADC," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International* (San Francisco, CA), 190–192. doi: 10.1109/ISSCC.2011.5746277

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 23 February 2014; accepted: 26 June 2014; published online: 22 July 2014.
Citation: Mayr CG, Partzsch J, Noack M and Schüffny R (2014) Configurable analog-digital conversion using the neural engineering framework. Front. Neurosci. ***8****:201. doi: 10.3389/fnins.2014.00201*
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience.

frontiers in NEUROSCIENCE

ORIGINAL RESEARCH ARTICLE
published: 22 July 2014
doi: 10.3389/fnins.2014.00205

Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array

Sukru B. Eryilmaz[1]*, Duygu Kuzum[2], Rakesh Jeyasingh[1], SangBum Kim[3], Matthew BrightSky[3], Chung Lam[3] and H.-S. Philip Wong[1]

[1] Department of Electrical Engineering, Stanford University, Stanford, CA, USA
[2] Department of Bioengineering, University of Pennsylvania, Philadelphia, PA, USA
[3] IBM Research, T. J. Watson Research Center, Yorktown Heights, NY, USA

Edited by:
Themis Prodromakis, University of Southampton, UK

Reviewed by:
Davide Badoni, University Tor Vergata, Italy
Omid Kavehei, University of Melbourne, Australia

***Correspondence:**
Sukru B. Eryilmaz, Department of Electrical Engineering and Center for Integrated Systems, Paul G. Allen B113X, 420 Via Palou, Stanford University, Stanford, CA 94305-4075, USA
e-mail: eryilmaz@stanford.edu

Recent advances in neuroscience together with nanoscale electronic device technology have resulted in huge interests in realizing brain-like computing hardwares using emerging nanoscale memory devices as synaptic elements. Although there has been experimental work that demonstrated the operation of nanoscale synaptic element at the single device level, network level studies have been limited to simulations. In this work, we demonstrate, using experiments, array level associative learning using phase change synaptic devices connected in a grid like configuration similar to the organization of the biological brain. Implementing Hebbian learning with phase change memory cells, the synaptic grid was able to store presented patterns and recall missing patterns in an associative brain-like fashion. We found that the system is robust to device variations, and large variations in cell resistance states can be accommodated by increasing the number of training epochs. We illustrated the tradeoff between variation tolerance of the network and the overall energy consumption, and found that energy consumption is decreased significantly for lower variation tolerance.

Keywords: phase change memory, synaptic device, neuromorphic computing, cognitive computing, device variation, associative learning, neural network, spike-timing-dependent-plasticity

INTRODUCTION

Historical improvements in cost and performance of CMOS technology have relied on transistor scaling for decades. However, CMOS transistor scaling has started reaching its physical as well as economic limits (Radack and Zolper, 2008). Further scaling may prevent reliable binary operation of CMOS devices. As devices are scaled down, device to device as well as cycle to cycle variations increase (Frank et al., 2001). Conventional digital logic based architectures cannot handle large variations as they are based on deterministic operation of devices; and extra circuitry aimed at mitigating these variations results in a huge overhead, increasing the cost significantly. In addition, increase in leakage current and hence the energy consumption as a result of further scaling imply that unabated scaling of transistor size is not the optimal solution for further performance increases (Frank et al., 2001). Furthermore, conventional information processing systems based on the von Neumann architecture have a performance bottleneck due to memory and processor being separated by a data channel. The increasing performance gap in the memory hierarchy between the cache and nonvolatile storage devices limits the system performance in Von Neumann architectures (Hennessy et al., 2012). Hence, in order to continue the historical performance improvements in information processing technology, different concepts and architectures need to be explored. New architectures are highly desired especially for specific applications that involve computation with a large amount of data and variables, such as large-scale sensor networks, image reconstruction tools, molecular dynamics simulations or large scale brain simulations (Borwein and Borwein, 1987).

Massive parallelism, robustness, error-tolerant nature, and energy efficiency of the human brain suggest a great source of inspiration for a non-conventional information processing paradigm which can potentially enable significant gains beyond scaling in CMOS technology and break the von Neumann bottleneck in conventional architectures (Mead, 1990; Poon and Zhou, 2011; Le et al., 2012). Synaptic electronics is an emerging field of research aiming to realize electronic systems that emulate the computational energy-efficiency and fault tolerance of the biological brain in a compact space (Kuzum et al., 2013). Since brain-inspired systems are inherently fault tolerant and based on information processing in a probabilistic fashion, they are well-suited for applications such as pattern recognition which operates on large amounts of imprecise input from the environment (Le et al., 2012). One approach to brain-like computation has been the development of software algorithms executed by supercomputers. However, since these have been executed on conventional architectures, they have not come close to the human brain in terms of performance and efficiency. For instance, IBM team has used the Blue Gene supercomputer for cortical simulations at the complexity of a cat brain (Preissl et al., 2012). Although this is a multi-core architecture, it is still nowhere close to the human brain in terms of parallelism, even though it already requires large

amount of resources: 144 TB of memory and 147,456 microprocessors, and consumes 1.4 mW of power overall (as opposed to approximately 20 W consumed in biological brain in humans) (Preissl et al., 2012). Another approach is to realize brain-like parallelism in hardware instead of programming conventional systems by software. Typically, the number of synapses (connection nodes between neurons) are much larger than number of neurons in a neural network, making synaptic device the most crucial element of the system in terms of area footprint and energy consumption to realize brain-like computing systems on hardware (Drachman, 2005). CMOS implementations of smaller scale physical neural networks on a specialized hardware have been previously demonstrated (Indiveri et al., 2006). The large area occupied by CMOS synapses limits the scale of the brain-like system that can be realized with these approaches. For instance, the synaptic element in Merolla et al. (2011) is an 8-transistor SRAM cell, with an area of $3.2 \times 3.2\,\mu m$ using a 45 nm CMOS technology. This area-inefficient synaptic element makes it impractical to scale up the system. Implementing synaptic functionality in a much more compact space, such as on the order of few tens of nanometers, would be useful to build a more compact intelligent architecture, besides potentially being more power efficient. Such a compact synaptic device is especially required when the goal is to upscale the system to the scale of human brain. In recent years, different types of emerging nanoscale nonvolatile memory devices, including phase change memory (PCM) (Kuzum et al., 2011; Bichler et al., 2012; Suri et al., 2012), resistive switching memory (RRAM) (Xia et al., 2009; Chang et al., 2011; Seo et al., 2011; Yu et al., 2011, 2013; Yang et al., 2012) and conductive bridge memory (CBRAM) (Jo et al., 2010; Ohno et al., 2011), have been proposed for implementing the synaptic element in a compact space. Such devices, which can be scaled to nanometer dimensions, would enable realization of highly dense synaptic arrays approaching human scale implementation of brain emulators or intelligent systems on hardware, owing to their small feature sizes. Among these different types of emerging memory devices, phase change memory has the advantage of being a more mature technology. In addition, phase change memory has excellent scalability. In fact, phase change material has shown switching behavior down to 2 nm size (Liang et al., 2012). Phase change memory arrays fabricated in 3-dimension have been demonstrated as an alternative approach for high density memory (Kinoshita et al., 2012). Functional arrays of phase change memory cells have already been demonstrated in 20 nm and other technology nodes (Servalli, 2009; Kang et al., 2011). Hence, it is possible to build a hybrid brain-like system using nanoscale synaptic devices using phase change memory integrated with CMOS neurons.

The main characteristic of PCM that makes it a good candidate as a synaptic device is its capability for being programmed to intermediate resistance states between high and low resistance values, or gradual programming (Kuzum et al., 2011). As illustrated by Kuzum et al., the ability to program a PCM in 1% gray-scale conductance levels enables the PCM to emulate the spike-timing-dependent plasticity (STDP) in synaptic strength in hippocampal synapses. Furthermore, the crossbar architecture used in most memory array configurations is actually analogous to grid-like connectivity of brain fibers in human brain (Wedeen et al., 2012).

The low resistance state of PCM is called the SET state and transition from the high resistance state to the low resistance state is called SET. High resistance state of PCM is called the RESET state and transition from low resistance state to the high resistance state is called RESET. Applying appropriate voltage pulses create intermediate resistance states between the fully SET state and the fully RESET state in a phase change memory device (Kuzum et al., 2011). This is similar to gradual weight change in biological synapses, where the synaptic weight is modified in accordance with relative arrival timing of the spikes from pre and post-neurons. This is called spike timing dependent plasticity (STDP), and is thought to be one of the fundamental learning rules in hippocampal synapses (Bi and Poo, 1998). Using this property of phase change devices as well as similar characteristics of other emerging memory devices mentioned above, network level learning studies have been done (Pershin and Di Ventra, 2010, 2011; Bichler et al., 2012; Alibart et al., 2013; Kaneko et al., 2013; Yu et al., 2013). However, many of these works studying nanoscale synaptic devices on network level have been limited to simulations, and experimental works either have used few number of synapses or lack a thorough analysis of important network parameters (Pershin and Di Ventra, 2010; Alibart et al., 2013; Kaneko et al., 2013). Recently, we presented preliminary findings of hardware demonstration of a synaptic grid using phase change memory devices as synaptic connections (Eryilmaz et al., 2013). In this work, we present a detailed description of the algorithm and signaling scheme used, and additionally present a thorough analysis of the tradeoff between the power consumption, the number of iterations required, and the device resistance variation. We experimentally study the effects of resistance variation on learning performance in the system level. We find that larger variations can be tolerated by increasing the number of learning epochs, but this comes with increased overall energy consumption, resulting in a trade-off between variation tolerance, energy consumption, and speed of the network.

PHASE CHANGE MEMORY CELL ARRAY FOR SYNAPTIC OPERATION

Phase Change Memory (PCM) cells used in the experiment are mushroom type cells, which means the heater material, bottom electrode (BE), phase change material, and the top electrode (TE) are stacked on top of each other, respectively (Wong et al., 2010). The 10-by-10 memory array used in the experiments consists of 100 memory cells. These cells are connected in a crossbar fashion as illustrated in **Figure 1A**. Each memory cell consist of a PCM element in series with a selection transistor. The circuit schematic of a memory cell is shown in **Figure 1A**, and a cross section of a memory cell is shown in **Figure 1B**, together with the optical microscope image of the memory chip used. The cells can be accessed through bitline (BL) and wordline (WL) nodes. Each wordline is connected to the gates of selection transistors of 10 memory cells, and each bitline is connected to the top electrode of the PCM element of 10 memory cells. Overall, there are 10 WL and 10 BL nodes in the array. Note that the bottom electrode of a PCM element within a cell is connected to the selection transistor

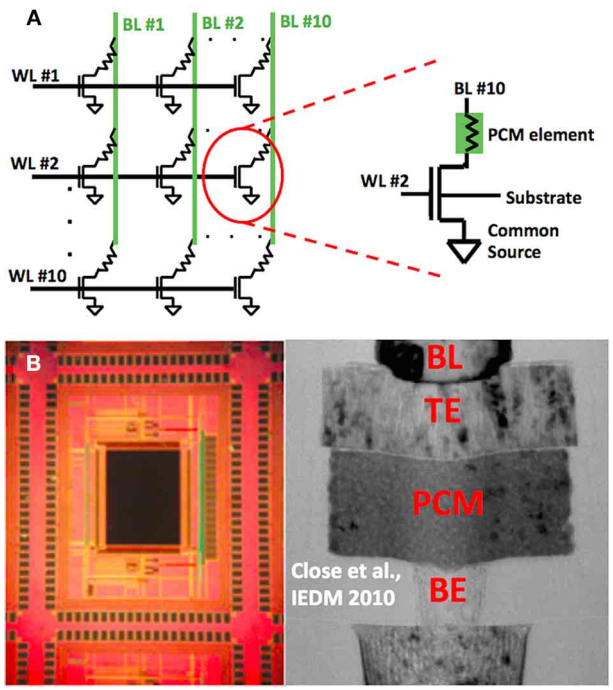


FIGURE 1 | (A) Schematic of 10 × 10 phase change memory (PCM) cell array is shown on the left. Resistances connected in series with the selection transistors represent PCM element. The figure on the right shows the complete schematic of a single memory cell. This particular cell can be accessed by applying appropriate biases at WL #2 and BL #10. Substrate and common source terminals are grounded during the experiment. **(B)** Optical microscope image of memory cell array located on the memory chip is shown on the left. TEM image of a single memory cell is appended to the right hand side. Mushroom type cell structure can be seen by observing the bitline (BL), top electrode (TE), phase change material (PCM) and bottom electrode (BE) stack (BE) stack. TEM image is reprinted with permission from Close et al. (2010) Copyright 2010 IEEE. TEM image is a representative figure for 90 nm node mushroom PCM cell, and PCM cells in the array in this paper are 180 nm node with the same device structure.

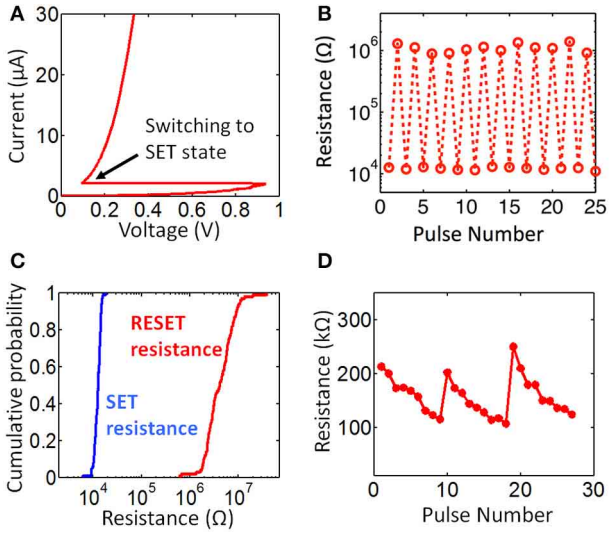


FIGURE 2 | Electrical characterization of memory cells. (A) shows the DC switching characteristics of a single memory cell arbitrarily selected from the array. Switching behavior can be observed when there is 2 μA of current through the memory cell. Binary switching cycles are shown in **(B)**. SET pulse is applied at odd numbers of measurement (pulse #1, 3, . . .) and RESET pulse is applied at even numbers of measurement. The plot shows the measured resistance of the memory cell right after the programming pulse is applied. Array level binary resistance distribution is shown in **(C)**. Resistance window for binary operation is larger than 10 k. Gradual resistance change in a single cell is shown in **(D)**. This plot is obtained by applying gradual SET pulses right after the cell is abruptly programmed to RESET state. The plot shows 3 cycles of this measurement.

of that cell. Each cell is associated with a unique (WL, BL) pair, hence each cell can be accessed by applying bias to the corresponding BL and WL nodes, as shown in **Figure 1A**. The device fabrication as well as retention and endurance characteristics of memory cells in the array are given in detail elsewhere (Close et al., 2010).

SET programming of a memory cell is achieved by applying a long (from a few hundred ns to few μs) current pulse through the PCM element to crystallize the phase change material in the PCM via Joule heating. In a gradual SET programming, depending on the amplitude of the current pulse, resistance of the PCM reduces for a certain amount, rather than going directly into the lowest resistance (fully SET) state (see **Figure 2D**). RESET (high resistance) programming is achieved by amorphizing the phase change material of the memory cell by applying a larger current pulse with a very sharp (2–10 ns fall time) falling edge. A large amplitude of current pulse results in melting of PCM material through Joule heating, the sharp falling edge quenches the cell, without allowing time for the phase change material to go into the more stable crystalline state, leaving it in the amorphous state. The amount of resistance increase for gradual RESET can be controlled either by changing the falling edge width of the current pulse or by changing the current pulse amplitude (Kang et al., 2008; Mantegazza et al., 2010). Typical DC switching characteristics of a single device arbitrarily chosen from an array are shown in **Figure 2A**. For DC switching characterization, 3.3 V is applied at WL of a single cell and BL node is swept from 0 V up to the switching threshold. The measurement result in **Figure 2A** shows that switching threshold for one of the cells in a fully RESET state is around 0.8 V, and the current when switching occurs is 2 μA. Note that these values can vary across the memory array due to device to device variation. Set and reset pulses with amplitudes of 1 V and 1.5 V and with (50 ns/300 ns/1 μs) and (20 ns/50 ns/5 ns) rise/width/fall time is applied at WL node, while BL node is held at 3.3 V during characterization of pulse switching in the memory cells. Pulse switching characteristics are shown in **Figure 2B**. This data is obtained by applying SET pulses for pulse #1,3,5. . . and RESET pulses for pulse #2,4,6. . . The same SET and RESET pulses are used for array level binary resistance characterization shown in **Figure 2C**. RESET resistance is distributed around 3 M ohms and SET resistance is distributed around 10 k ohms. For synaptic operation, gradual resistance change characteristics of memory cells are utilized. Specifically, our system utilizes gradual SET programmability of memory cells. To characterize gradual resistance change from the RESET state to the partially SET state, we apply

once a 1.1 V RESET pulse and then 9 SET pulses with 0.85 V amplitude. Gradual resistance change characteristics from RESET to SET for a single cell is shown in **Figure 2D** for a few cycles of gradual SET characterization. This gives us around 9 resistance levels between low and high resistance state. Although the energy consumption for gradual SET is lower than gradual RESET, variability is larger for gradual SET since gradual SET is probabilistic in nature (Braga et al., 2011). The reason behind this is the intrinsic stochasticity of the nucleation of crystalline clusters during gradual SET operation. The cycle-to-cycle variability is also observed in **Figure 2D** The same resistance levels are not accurately repeatable from cycle to cycle. Due to variability in gradual resistance change, multi-level-cell (MLC) memory applications use a write-and-verify technique since the data storage applications require deterministic binary resistance levels (Kang et al., 2008). However, massively-parallel brain-like architectures can tolerate such variations and do not require the use of write-and-verify that is needed to achieve an accurate resistance level. Hence, the variations observed in **Figure 2D** do not pose a problem for our purposes.

ARRAY LEVEL LEARNING

A fully-connected recurrent Hopfield network is employed for the learning experiments (**Figure 3A**) (Hertz et al., 1991). The Hopfield network consists of 100 synaptic devices and 10 recurrently connected neurons, as shown in **Figure 3A**. It is worth noting that in this architecture, all neurons are both input and output neurons. Integrate-and-fire neurons are implemented by computer control and memory cells serve as synaptic devices between neurons. **Figure 3A** illustrates how the network is constructed using the memory cell array. The input terminal of the i-th neuron is connected to BL #i, and output terminal of the i-th neuron is connected to WL #i, where $i = 1, 2, \ldots, 10$, i.e., neuron #1 input and output is connected to BL #1 and WL #1, respectively, and neuron #2 input and output is connected to BL #2 and WL #2, respectively, etc. (**Figure 3A**). Before the experiment, all synapses are programmed to the RESET state. A learning experiment consists of epochs during which synaptic weights are updated depending on firing neurons. After training, the pattern is presented again but with an incorrect pixel this time, and the incorrect pixel is expected to be recalled in the recall phase

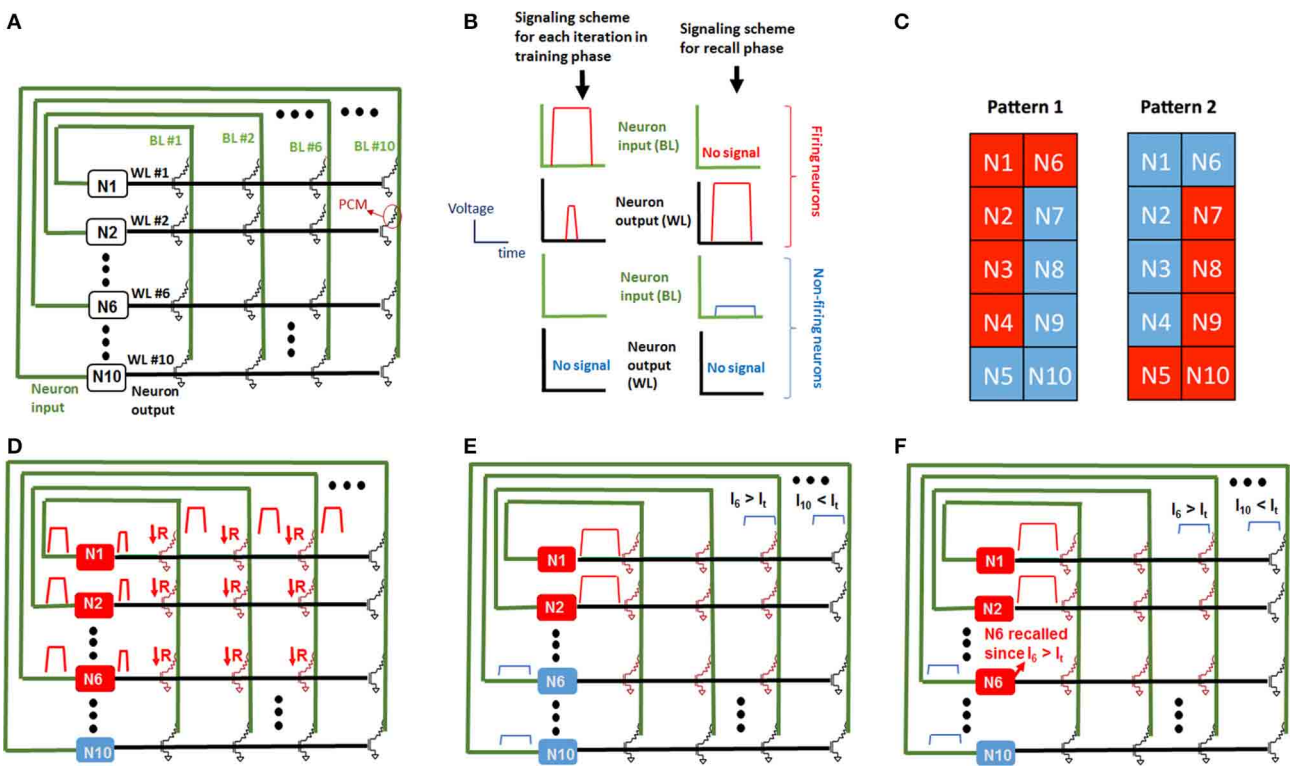


FIGURE 3 | Neural network realized and how it is implemented with the memory array is explained. (A) Shows the recurrently connected Hopfield network implemented in the learning experiment. Pulsing scheme during training as well as recall is shown in **(B)**. We train the network with two patterns as shown in **(C)**, where red pixels correspond to ON and blue pixels correspond to OFF. Numbers in pixels correspond to the neuron number associated with that pixel. During update phase shown in **(D)**, the resistance of synaptic elements connected to non-firing neurons do not change, since no pulse is applied at the WL node of non-firing neurons during update phase. The synaptic connections between firing neurons, however, are programmed by the pulses applied at the BL and WL of the corresponding memory cell. The pulse characteristics are predetermined for gradual SET programming of the memory cell, hence the resistance is reduced with an amount and the connection gets stronger. **(E)** During the read phase, a small amplitude voltage applied at the BL node of non-firing neurons sense the total current due to the synapses of that neuron connected to firing neurons, since a pulse applied at the output of the firing neurons turns the selection transistor on simultaneously. **(F)** In this example, during the recall phase, N1–N4 are presented with N6 OFF (not firing), but N6 is recalled since the input current of N6 is larger than the threshold.

after training is performed (**Figure 3B**). A complete pattern is presented during the training phase of an epoch, and an incomplete pattern with an incorrectly OFF pixel is presented during the recall phase. All patterns consist of 10 pixels, and each neuron is associated with a pixel. This mapping between pixels and neurons is shown in **Figure 3C** for two different patterns considered in this work. **Figure 3B** shows the pulsing scheme for firing and non-firing neurons in both update and recall phases. When a pattern is presented during a training phase, the neurons associated with ON (red pixels in **Figure 3C**) pixels are externally stimulated, hence they fire. As can be seen in **Figure 3D**, when a neuron spikes during the training phase, it applies programming pulses at its input (corresponding BL) and output (WL). This results in gradual SET programming of the synaptic device between those two firing neurons. For instance, when neuron 1 and neuron 2 fire, programming pulses are applied at WL1, WL2, BL1, and BL2, as defined in the pulsing scheme in **Figure 3B**. These pulses will result in a current going through PCM elements and hence gradual SET programming of memory cells that connect neuron 1 and neuron 1 (see **Figure 3D**). After training, the recall phase begins. During the recall phase, a pattern with an incorrectly OFF pixel is presented (**Figure 3E**). Again, the neurons associated with ON pixels during recall phase fire, and appropriate pulses are applied at the input and output of neurons as shown in the pulsing scheme in **Figure 3B**). Neurons associated with OFF pixels during recall phase do not fire. Note that there is a low amplitude pulse applied at the input of non-firing neurons during recall phase. This voltage pulse, together with the large amplitude voltage pulse applied at the firing neurons' output during recall phase, create an input current feeding into non-firing neurons. The amplitude of this current through a non-firing neuron is determined by the resistance values of synaptic connections between that neuron and the firing neurons. This input current of non-firing neurons during recall phase is analogous to membrane potential of biological neurons. In biological neurons, the postsynaptic current feeding into a neuron accumulates charge on capacitive membrane, forming a membrane potential. Typically, this is modeled by a time constant that is determined by membrane capacitance. In this experiment, neurons fire simultaneously during the recall phase, while at the same time the input current through the non-firing neurons is measured. Since the delays and timing properties of the neurons are not included in the neuron model, the membrane capacitance is not included in neurons. Hence, input current through a neuron is actually equivalent to membrane potential in our experiments. Note that in this paper, we will use the terms input current and membrane voltage interchangeably, due to the reasons explained above. The input current into a non-firing neuron during recall phase can be written as follows:

$$I_i = V_{read} \sum_{j \in F} \frac{1}{R_{ij}} \quad (1)$$

In Equation (1), I_i is the input current into the ith neuron where it is a non-firing neuron, F is the set of indices of firing neurons, R_{ij} is the resistance of synaptic element between bitline i and wordline j, and V_{read} is the read voltage at the input of non-firing neurons during recall phase (see **Figure 3B**), which is 0.1 V in our experiments. As **Figure 3B** shows, if a neuron is not associated with an OFF pixel at the beginning of the recall phase, it fires, and the reading voltage V_{read} at its input is 0, making its input 0.

If the input current through a non-firing neuron exceeds a threshold during the recall phase, then the neuron associated with the pixel fires, the complete pattern is recalled (**Figure 3F**). The membrane potential of neurons is set to 0 at the beginning of each epoch, hence it does not transfer to the next epoch. We define "missing pixel" as the pixel that is ON in the correct pattern used for training, but OFF in the input pattern during recall phase. Note that the pixel missing from the pattern in recall phase still fires in update phase during training, SET programming the corresponding memory cells between this neuron and other firing neurons. This results in a decrease in the resistance values between this missing pixel's neuron and other firing neurons (ON pixels) as shown in **Figure 3D**), increasing the input current of the missing pixel's neuron during the recall phase (**Figure 3F**). Hence, recall is expected to occur after a few epochs, at which point the membrane potential exceeds a pre-determined threshold. This learning scheme is a form of Hebbian learning, since the weights of synaptic connections between coactive neurons during training phase get stronger, due to reduced resistances of these synaptic connections. The time window that defines the firing of two neurons as being coactive is determined by the width of the pulse applied at the input of firing neurons during update phase, shown in **Figure 3B**. This time window is 100 μs in our experiments. As an illustration of the aforementioned learning process, two simple 10-pixel patterns are chosen to be learned. The two patterns of 10 pixels are shown in **Figure 3C**. The network is first trained with pattern 1 (on the left in **Figure 3C**), and then pattern 2 (on the right in **Figure 3C**). During training with pattern 1, until the pattern is recalled, the complete pattern is presented in training phase and the pattern with pixel 6 missing is presented during recall phase. After pattern 1 is recalled, the same procedure is performed for pattern 2, this time with pixel 5 missing in the recall phases of epochs. This experiment is performed for 4 cases, each corresponding to different initial resistance variations across the array. Initial variation here refers to the variation after all cells are programmed to RESET before learning experiment begins. Different initial variation values are obtained by individually programming the memory cells in different arrays. The evolution of synaptic weights is shown in **Figure 4** during the experiment for the case where the initial variation is 60%. Note that the synaptic weight map in **Figure 4** shows the normalized synaptic weights of each synaptic device. Each data point in this map shows the resistance of the synaptic device after the corresponding epoch divided by the initial RESET resistance (right before the experiment when all devices are RESET programmed as explained above) of that device. Hence the map does not include the variations of initial RESET state resistances across the array. The variation study is explained in the next section. As can be seen in **Figure 4**, after feeding each input pattern into the network, synapses between the ON neurons gradually get stronger (resistance decreases); after 11 epochs, patterns are recalled. The overall energy consumed in synaptic devices during this experiment is 52.8 nJ. This energy does not include the

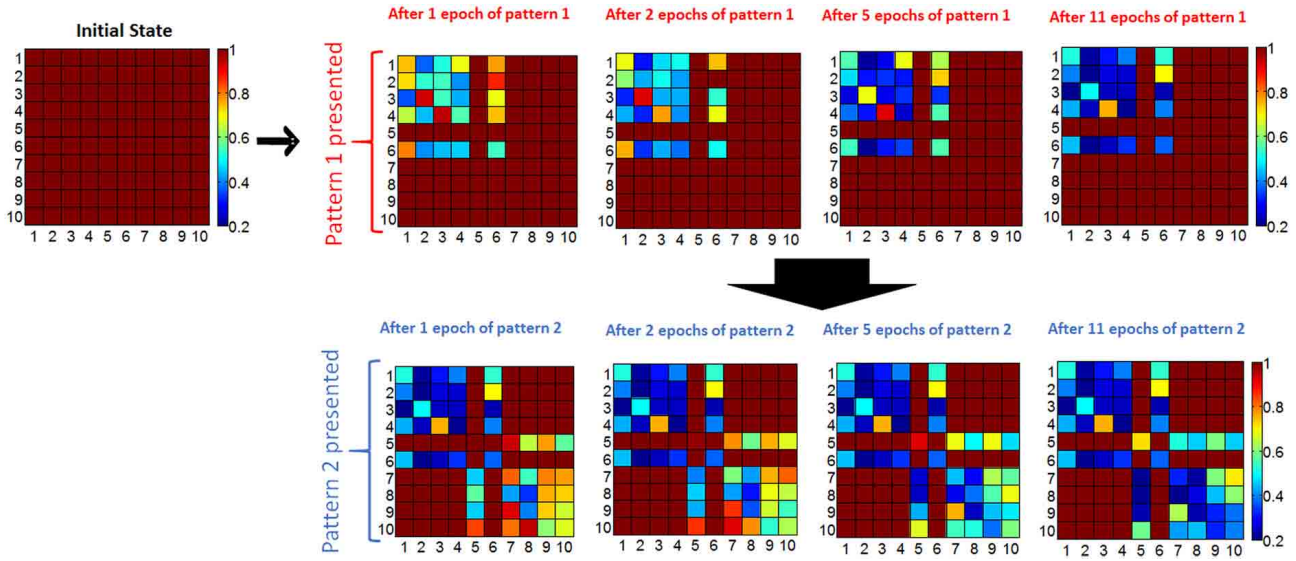


FIGURE 4 | Evolution of normalized resistance of synaptic devices is shown, for the 60% initial variation case. All normalized resistances are 1 initially since the normalized resistance map shows the current resistance of a synaptic device divided by its initial resistance. Note that the row and column numbers corresponds to BL and WL that connect the synaptic devices. For instance, the data shown in row #3 and column #6 is the normalized resistance of the memory cell that can be accessed by BL #6 and WL #3. First, pattern 1 is presented to the network. For pattern 1, ON neurons for the complete pattern during update phase are N1, N2, N3, N4, N6, and, and for the recall phase N6 is OFF and expected to be recalled (i.e., expected to fire) after training with a certain number of epochs. The gradual decrease in the normalized resistance of synaptic connections between firing neurons during the update phase can be observed. After 11 epochs, when recall phase is performed, OFF pixel #6 (neuron #6) is recalled (meaning neuron #6 fires in recall phase) , and then pattern 2 is presented for training. For pattern 2, the complete pattern is represented by N5, N7, N8, N9, N10; and N5 is missing in the recall phase.

energy consumed in the neurons and the wires, and is the energy consumed by the synaptic devices during training and recalling of pattern 1. Our measurements indicate that roughly 10% of this energy is consumed in phase change material, while around 90% is consumed in selection devices in our experiment. Note that the number of epochs and the overall energy consumed strongly depends on the choice for the threshold membrane potential of neurons. If threshold membrane potential is kept low, the number of epochs would be reduced, but a wrong pixel might fire (hence turn on) in the output of recall phase due to variations, hence recalling a wrong pattern. This is explained in detail in the next section.

EFFECT OF VARIATION ON LEARNING PERFORMANCE

Figure 5A shows the actual resistance map of synaptic connections after 11 epochs for the experiment above, along with the resistance distribution (on the left in **Figure 5**) when all the cells are in the RESET state before the experiment. As the synaptic connections evolve during training for two patterns, synapses between coactive neurons get stronger. Actual resistance maps in **Figure 5** also illustrate the resistance variation across the array when all cells are in RESET state before training. In our experiment, the neuron firing threshold is the important parameter that can be tuned to tolerate the variation. This threshold value has to be large enough so that a wrong pixel will not turn on in recall phase, but low enough to guarantee that the overall energy consumed is minimal and the missing pixel will actually turn on in recall phase, hence recalling the original pattern. To this end, the firing threshold of neurons is selected as follows:

$$I_{thr} = C \cdot \max_{N,i} \left(V_{read} \sum_{j \in N} \frac{1}{R_{ij}} \right) \quad (2)$$

In Equation (2), N is constrained to be a 4-element subset of the set {1,2,3,... 9,10}, and R_{ij} is the initial RESET resistance of the memory cell defined by bitline i and wordline j, and V_{read} is defined as in Equation (1). This equation means that the threshold current is a constant C times the largest input current that a neuron can possibly have in the recall phase, given the resistance values for each cell. The reason for considering 4-element subsets is because we are assuming 4 pixels are ON in the input during recall phase, and we want to make sure that the threshold is large enough to avoid firing of a neuron during recall phase when it is actually not ON in the true pattern. In its current form, this scheme might not be successful when different number of pixels are missing, for example, when three pixels are ON in recall phase while 5 pixels are ON in the actual pattern. This generalization can be made by allowing negative weights; equivalently using 2-PCM synapse suggested in Bichler et al. (2012), or adaptive threshold method suggested in Hertz et al. (1991). The requirement that $C > 1$ guarantees that during the training, the wrong pixel will not be recalled at any epoch. This is because the resistance of the synaptic connections between an arbitrary OFF pixel in the original

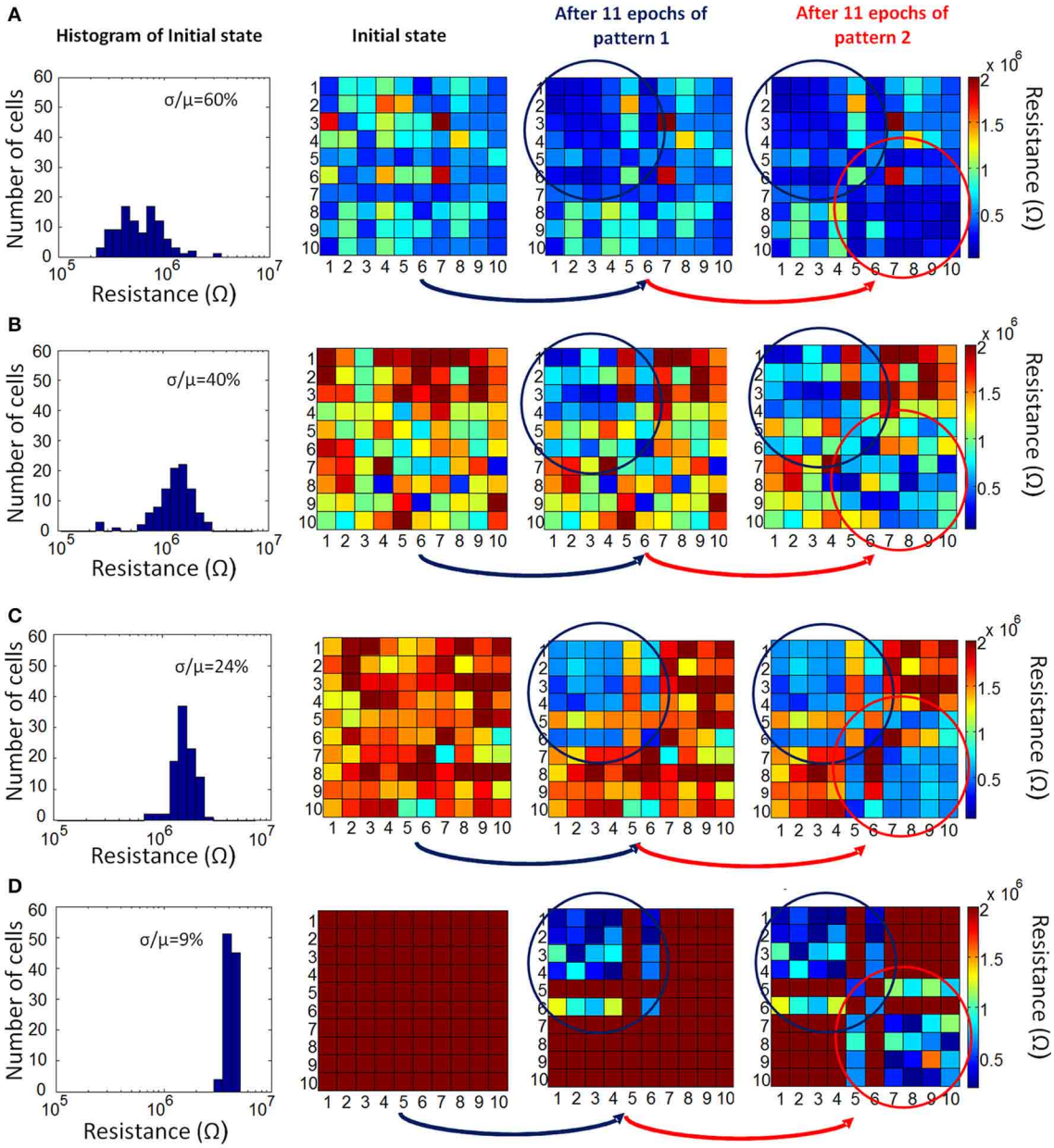


FIGURE 5 | Evolution of actual resistance of synaptic devices is shown for four different initial resistance variation cases: (A) 60%, (B) 40%, (C) 24% and (D) 9%. The representation of synaptic devices in these resistance maps are the same as in **Figure 4**, but this time the resistance values are not normalized. The variations across the memory cell arrays are apparent here. Synaptic devices between firing neurons during training get stronger (i.e., are driven to lower resistance values). As the initial variation reduces, the difference in resistance values between potentiated synapses and the synapses that remain unchanged becomes more pronounced.

pattern and other neurons do not decrease, as the OFF pixels do not fire during training. We choose $C = 2$ for our experiments. Choosing $C = 2$ also allows us, without requiring negative synaptic weights, to generalize recall to some extent for inputs with incorrectly ON pixels, in addition to incorrectly OFF pixels as given in our example. This idea is similar to adaptive threshold method in Hertz et al. (1991), where instead of using negative weights, neuron threshold is increased while keeping the weights positive. Observe that as the variation increases, the low-resistance tail of the initial RESET resistance distribution (leftmost histograms in **Figures 5A–D**) extends toward lower resistance values. This results in a decrease in minimum resistance values, as can be seen in histograms in **Figures 5A–D**). Hence, maximum neuron input current with 4 neurons firing increases. This increases the max term in Equation (2), hence a higher number of epochs is needed to recall the missing pixel for larger

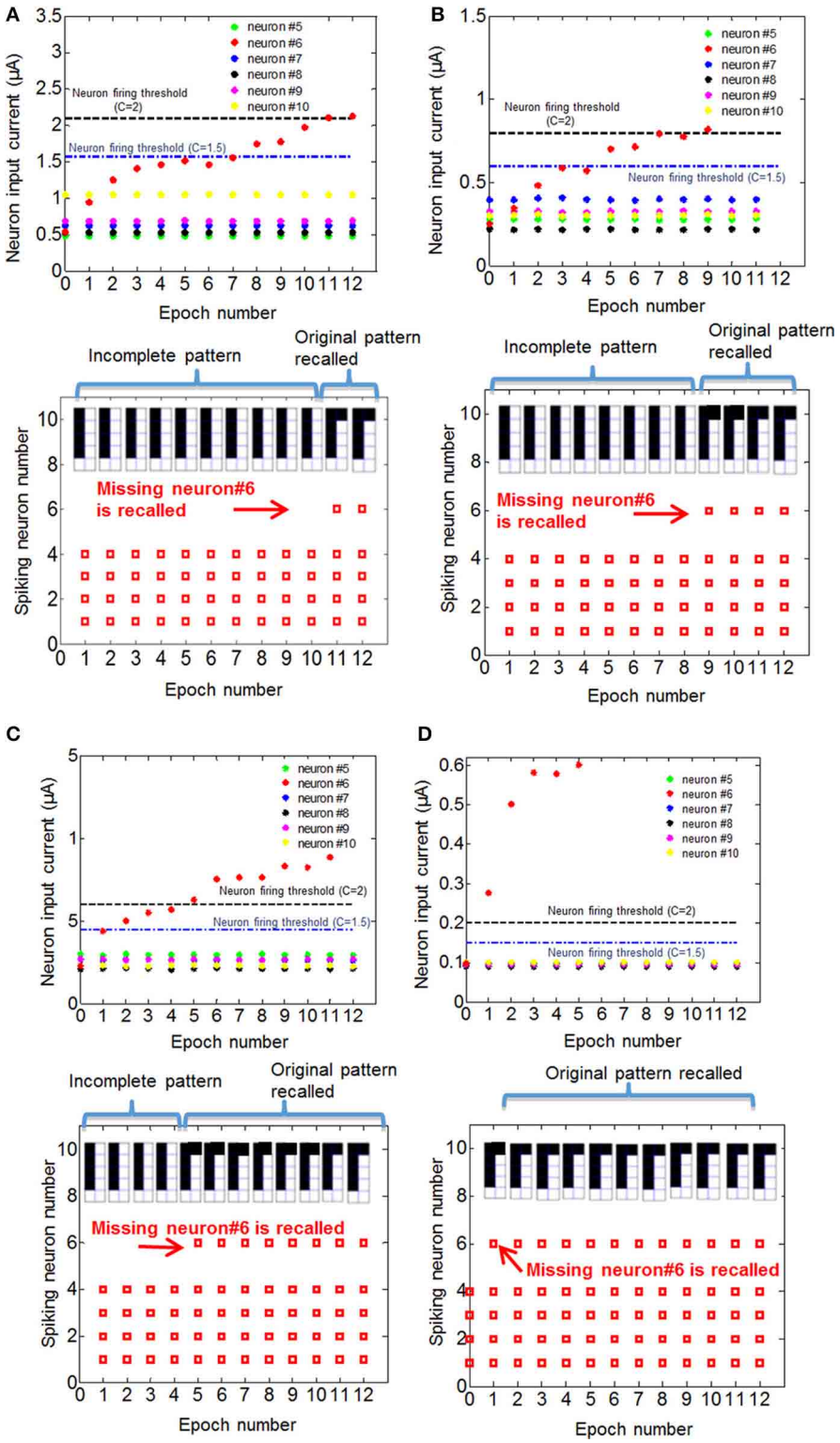


FIGURE 6 | Recall of the missing pixel for training with pattern 1 for four different initial variation cases, (A) 60%, (B) 40%, (C) 24%, and (D) 9%, are shown. For each case, top figures show what the input current of neurons that do not fire would be if the recall is performed after the corresponding number of epochs, and bottom figures show the neurons that fires if the recall was performed after the corresponding number of epochs for $C = 2$ (see the text for details about parameter **C**). Different threshold levels for $C = 1.5$ and $C = 2$ cases are shown in the top figures. When the input current exceeds the threshold after a certain number of epochs, the missing pixel N6 fires. For $C = 2$, the number of epochs after which N6 fires in each case is 11 (60% variation), 9 (40% variation), 5 (24% variation) and 1 (9% variation).

variation. The resistance maps for other variation cases are shown in **Figures 5B–D**. We can see that as initial variation reduces, the same number of epochs yields a more pronounced overall difference between the weights that get stronger versus the weights that do not change, as illustrated in **Figure 5**. The evolution of the membrane potential with the number of epochs for different variation cases are shown in **Figure 6**. While it takes 11 epochs to recall a pattern when there is 60% initial variation, only one epoch is sufficient in our case when the initial resistance variation is 9%. It is worth mentioning that we have negligible variation in read voltage during our experiment, since the reading of memory cell resistances is performed with electronic equipment. When this synaptic grid is integrated with actual CMOS neurons, however, it is expected to have some variation in read voltage, which results in variation in the input current of neurons. This variation in input current might cause variations in the number of epochs needed for training. We can observe from **Figure 6** that while it takes 3% input current variation (hence read voltage variation) to change the number of epochs needed for 60% variation case (**Figure 6A**), it takes 40% variation in read voltage to change the number of training epochs required for 9% initial variation case (**Figure 6D**). This is because as the number of epochs increases, resistances of programmed synapses begin to converge to low resistance values. To minimize the effect of read voltage variation, properties of synaptic device as well as pulsing scheme during training should be carefully chosen, considering the read voltage variation of CMOS neuron circuit. The increase in the required number of epochs to recall the pattern results in a higher overall energy consumption. Overall energy consumption for 9% initial resistance variation case is 4.8 nJ, whereas it is 52.8 nJ for 60% initial variation case. **Figure 7** illustrates the dependence of energy consumption and number of epochs needed on initial resistance variation. As can be seen in **Figure 7**, there is a clear reduction in the overall energy consumption as initial resistance variation goes down. Note that these energy values represent only the energy consumed in the synaptic devices for training and recall phases for pattern 1. They do not include the energy consumed in the wires or the neurons. Energy consumption in the wires can be a substantial part of the overall energy consumption for a large array (Kuzum et al., 2012). It is also worth noting that since the time scale between the epochs in these experiments is on the order of seconds, we did not observe any effects of drift in our measurements, which would require a timescale of μs or ms to observe (Karpov et al., 2007).

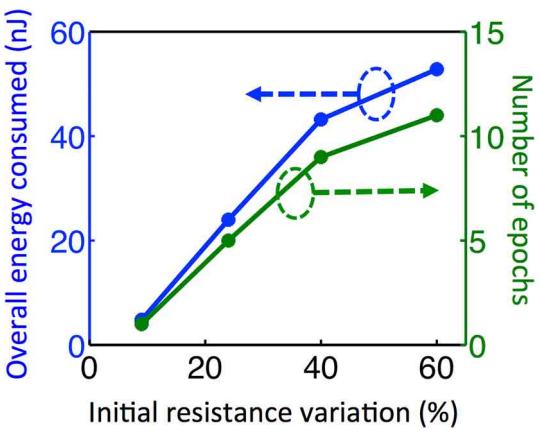


FIGURE 7 | The same experiment is repeated for different initial variation cases. In order to guide the eye, dashed arrows and circles indicate which curves correspond to which axis. For four different initial variation cases, the plot shows the total number of epochs required for training as well as overall energy consumed in the synaptic devices during training and recall phases for pattern 1. As the variation increases, larger firing threshold is required for neurons. This increases the number of epochs and energy consumption required for training.

CONCLUSION

We report brain-like learning in hardware using a crossbar array of phase change synaptic devices. We demonstrated in hardware experiments that synaptic network can implement robust pattern recognition through brain-like learning. Test patterns were shown to be stored and recalled associatively via Hebbian plasticity in a manner similar to the biological brain. Increasing the number of training epochs provides a better tolerance for initial resistance variations, at the cost of increased energy consumption. Demonstration of robust brain-inspired learning in a small-scale synaptic array is a significant milestone toward building large-scale computation systems with brain-level computational efficiency.

METHODS

The memory cell array was probed using a 25 × 1 probe card which is connected to a switch matrix consisting of two cards, each providing a 4 × 12 matrix (see **Supplementary Figure 1**). The probe card contacts 25 pads on the wafer that has the memory arrays. These 25 pads consist of 10 bitlines, 10 wordlines, 1 common source terminal, 1 substrate terminal, and 3 floating terminals. Switch matrix is connected to Agilent 4156C semiconductor analyzer to perform DC measurements and Agilent 81110 pulse generator for pulse measurements. All these equipment is controlled by a Labview program on a separate computer. This program allows us to switch between cells on the array automatically and applying custom signals from semiconductor analyzer or the pulse generator to the desired cell. In all the measurements, resistance of the memory cell is measured by applying 0.1 V read voltage at the bitline and 3.3 V at the wordline. The current (I) through the cell is measured and resistance is obtained by R = 0.1 V/I. DC switching measurement in **Figure 2A** is obtained from an arbitrarily selected cell on the array. For this particular measurement, current through the device is swept. For binary switching measurement in **Figure 2B**, alternating SET pulses (1 V amplitude, 50 ns/300 ns/1 μs rise/width/fall time) and RESET pulses (1.5 V amplitude, 5 ns/50 ns/5 ns rise/width/fall time) are applied by pulse generator. For the measurement in **Figure 2C**, the same SET and RESET pulses are applied at each 100 cells in an array. The gradual SET characteristics in **Figure 2D** is obtained by applying 1.1 V RESET pulse once and then 0.85 V gradual SET pulse 9 times. This cycle is repeated for a few times to obtain the result in **Figure 2D**. During learning experiment, the initial RESET programming of the cells before learning experiment starts was done by applying a RESET pulse (1.5 V amplitude,

5 ns/50 ns/5 ns rise/width/fall time) at every cell within the array. The energy consumed during gradual SET programming of synaptic connections in update phases is extracted by measuring the current through the devices during programming. Fraction of energy consumed in phase change material and in selection transistor is extracted by measuring individual transistor characteristics separately, as well as by current sweep measurements in PCM cells.

ACKNOWLEDGMENTS

This work is supported in part by Systems on Nanoscale Information Fabrics (SONIC) Center, one of six centers of Semiconductor Technology Advanced Research Network (STARnet), a Semiconductor Research Corporation (SRC) program sponsored by Microelectronics Advanced Research Corporation (MARCO) and Defense Advanced Research Projects Agency (DARPA), the NSF Expeditions in Computing (award 1317470), and the member companies of the Stanford Non-Volatile Memory Technology Research Initiative (NMTRI).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://www.frontiersin.org/journal/10.3389/fnins.2014.00205/abstract

Supplementary Figure 1 | Measurement setup used in experiments. Probe card that directly probes pads on memory chip are connected to switch matrix. Setup is controlled by computer program.

REFERENCES

Alibart, F., Zamanidoost, E., and Strukov, D. B. (2013). Pattern classification by memristive crossbar circuits using ex situ and in situ training. *Nat. Commun.* 4, 2072. doi: 10.1038/ncomms3072

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

Bichler, O., Suri, M., Querlioz, D., Vuillaume, D., DeSalvo, B., and Gamrat, C. (2012). Visual pattern extraction using energy-efficient "2-PCM synapse" neuromorphic architecture. *IEEE Trans. Electron Device* 59, 2206–2214. doi: 10.1109/TED.2012.2197951

Borwein, J. M., and Borwein, P. B. (1987). *Pi and the AGM: a Study in Analytic Number Theory and Computational Complexity*. New York, NY: John Wiley & Sons, Inc.

Braga, S., Cabrini, A., and Torelli, G. (2011). Experimental analysis of partial-SET state stability in phase-change memories. *IEEE Trans. Electron Device* 58, 517–522. doi: 10.1109/TED.2010.2090157

Chang, T., Jo, S.-H., and Lu, W. (2011). Short-term memory to long-term memory transition in a nanoscale memristor. *ACS Nano* 5, 7669–7676. doi: 10.1021/nn202983n

Close, G. F., Frey, U., Breitwisch, M., Lung, H. L., Lam, C., Hagleitner, C., et al. (2010). "Device, circuit and system-level analysis of noise in multi-bit phase change memory," in *IEEE International Electron Devices Meeting* (San Francisco, CA), 29.5.1–29.5.4.

Drachman, D. (2005). Do we have brain to spare? *Neurology* 64, 2004–2005. doi: 10.1212/01.WNL.0000166914.38327.BB

Eryilmaz, S. B., Kuzum, D., Jeyasingh, R. G. D., Kim, S., BrightSky, M., Lam, C., et al. (2013). "Experimental demonstration of array-level learning with phase change synaptic devices," in *IEEE International Electron Devices Meeting* (Washington, DC), 25.5.1–25.5.4.

Frank, D. J., Dennard, R. H., Nowak, E., Solomon, P. M., Taur, Y., and Wong, H.-S. (2001). Device scaling limits of Si MOSFETs and their application dependencies. *Proc. IEEE* 89, 259–288. doi: 10.1109/5.915374

Hennessy, J. L., Patterson, D. A., and Asanoviæ, K. (2012). *Computer Architecture: a Quantitative Approach*, *5th Edn.* Amsterdam: Morgan Kaufmann/Elsevier.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation.* Redwood City, CA: Addison-Wesley Publishing Company.

Indiveri, G., Chicca, E., and Douglas, R. (2006). A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Trans. Neural Netw.* 17, 211–221. doi: 10.1109/TNN.2005.860850

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301. doi: 10.1021/nl904092h

Kaneko, Y., Nishitami, Y., Ueda, M., and Tsujimura, A. (2013). "Neural network based on a three-terminal ferroelectric memristor to enable on-chip pattern recognition," in *Symposium on VLSI Technology* (Kyoto), 238–239.

Kang, D.-H., Lee, J.-H., Kong, J. H., Ha, D., Yu, J., Um, C. Y., et al. (2008). "Two-bit cell operation in diode-switch phase change memory cells with 90 nm technology," in *Symposium on VLSI Technology* (Honolulu, HI), 98–99.

Kang, M. J., Park, T. J., Kwon, Y. W., Ahn, D. H., Kang, Y. S., Jeong, H., et al. (2011). "PRAM cell technology and characterization in 20nm node size," in *IEEE International Electron Devices Meeting* (Washington, DC), 39–42.

Karpov, I. V., Mitra, M., Kau, D., Spadini, G., Kryukov, Y. A., and Karpov, V. G. (2007). Fundamental drift of parameters in chalcogenide phase change memory. *J. Appl. Phys.* 102, 124503–124506. doi: 10.1063/1.2825650

Kinoshita, M., Sasago, Y., Minemura, H., Anzai, Y., Tai, M., Fujisaki, Y., et al. (2012). "Scalable 3-D vertical chain-cell-type phase change memory with 4f2 poly-Si diodes," in *Symposium on VLSI Technology* (Honolulu, HI), 35–36.

Kuzum, D., Jeyasingh, R. G. D., Lee, B., and Wong, H.-S. P. (2011). Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Lett.* 12, 2179–2186. doi: 10.1021/nl201040y

Kuzum, D., Jeyasingh, R. G. D., Yu, S., and Wong, H.-S. P. (2012). Low-energy robust neuromorphic computation using synaptic devices. *IEEE Trans. Electron Device* 59, 3489–3494. doi: 10.1109/TED.2012.2217146

Kuzum, D., Yu, S., and Wong, H.-S. P. (2013). Synaptic electronics: materials, devices and applications. *Nanotechnology* 24:382001. doi: 10.1088/0957-4484/24/38/382001

Le, Q. V., Ranzato, M. A., Monga, R., Devin, M., Chen, K., Corrado, G. S., et al. (2012). "Building high-level features using large scale unsupervised learning," in *Proceedings of the 29th International Conference on Machine Learning* (Edinburgh).

Liang, J., Jeyasingh, R. G. D., Chen, H.-Y., and Wong, H.-S. P. (2012). An ultra-low reset current cross-point phase change memory with carbon nanotube electrodes. *IEEE Trans. Electron Device* 59, 1155–1163. doi: 10.1109/TED.2012.2184542

Mantegazza, D., Ielmini, D., Pirovano, A., and Lacaita, A. L. (2010). Incomplete filament crystallization during SET operation in PCM cells. *IEEE Electron Device Lett.* 31, 341–343. doi: 10.1109/LED.2010.2042273

Mead, C. (1990). Neuromorphic electronic systems. *Proc. IEEE* 78, 1629–1636. doi: 10.1109/5.58356

Merolla, P., Arthur, J., Akopyan, F., Imam, N., Manohar, R., and Modha, D. S. (2011). "A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45 nm," in *IEEE Custom Integrated Circuits Conference* (San Jose, CA).

Ohno, T., Hasegawa, T., Tsuruoka, T., Terabe, K., Gimzewski, J. K., and Aono, M. (2011). Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. *Nat. Mater.* 10, 591–595. doi: 10.1038/nmat3054

Pershin, Y. V., and Di Ventra, M. (2010). Experimental demonstration of associative memory with memristive neural networks. *Neural Netw.* 23, 881–886. doi: 10.1016/j.neunet.2010.05.001

Pershin, Y. V., and Di Ventra, M. (2011). Solving mazes with memristors: a massively parallel approach. *Phys. Rev. E* 84, 046703. doi: 10.1103/PhysRevE.84.046703

Poon, C.-S., and Zhou, K. (2011). Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities. *Front. Neurosci.* 5:108. doi: 10.3389/fnins.2011.00108

Preissl, R., Wong, T. M., Datta, P., Flickner, M., Singh, R., Esser, S. K., et al. (2012). "Compass: a scalable simulator for an architecture for cognitive computing," in *ACM/IEEE Conference High Performance Networking Computing, EEE, Storage and Analysis* (Salt Lake City, UT).

Radack, D. J., and Zolper, J. C. (2008). A future of integrated electronics: moving off the roadmap. *Proc. IEEE* 96, 198–200. doi: 10.1109/JPROC.2007.911049

Seo, K., Kim, I., Jung, S., Jo, M., Park, S., Park, J., et al. (2011). Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide

bilayer resistive switching device. *Nanotechnology* 22:254023. doi: 10.1088/0957-4484/22/25/254023

Servalli, G. (2009). "A 45nm generation phase change memory technology," in *IEEE International Electron Devices Meeting* (Baltimore, MD), 1–4.

Suri, M., Bichler, O., Querlioz, D., Traore, B., Cueto, O., Perniola, L., et al. (2012). Physical aspects of low power synapses based on phase change memory devices. *J. Appl. Phys.* 112, 054904. doi: 10.1063/1.4749411

Wedeen, V. J., Rosene, D. L., Wang, R., Dai, G., Mortazavi, F., Hagmann, P., et al. (2012). The geometric structure of the brain fiber pathways. *Science* 335, 1628–1634. doi: 10.1126/science.1215280

Wong, H.-S. P., Raoux, S., Kim, S., Liang, J., Reifenberg, J. P., Rajendran, B., et al. (2010). Phase change memory. *Proc. IEEE* 98, 2201–2227. doi: 10.1109/JPROC.2010.2070050

Xia, Q., Robinett, W., Cumbie, M. W., Banerjee, N., Cardinali, T. J., Yang, J. J., et al. (2009). Memristor-CMOS hybrid integrated circuits for reconfigurable logic. *Nano Lett.* 9, 3640–3645. doi: 10.1021/nl901874j

Yang, R., Terabe, K., Liu, G., Tsuruoka, T., Hasegawa, T., Gimzewski, J. K., et al. (2012). On-demand nanodevice with electrical and neuromorphic multifunction realized by local ion migration. *ACS Nano* 6, 9515–9521. doi: 10.1021/nn302510e

Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., and Wong, H.-S. P. (2013). Stochastic learning in oxide binary synaptic device for neuromorphic computing. *Front. Neurosci.* 7:186. doi: 10.3389/fnins.2013.00186

Yu, S., Wu, Y., Jeyasingh, R. G. D., Kuzum, D., and Wong, H.-S. P. (2011). An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. *IEEE Trans. Electron Devices* 58, 2729–2737. doi: 10.1109/TED.2011.2147791

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 01 May 2014; accepted: 30 June 2014; published online: 22 July 2014.
Citation: Eryilmaz SB, Kuzum D, Jeyasingh R, Kim S, BrightSky M, Lam C and Wong H-SP (2014) Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array. Front. Neurosci. ***8****:205. doi: 10.3389/fnins.2014.00205*
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: info@frontiersin.org | +41 21 510 17 00

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership