Device-aware Circuit Design for Robust Memristive Neuromorphic Systems with STDP-based Learning

SAGARVARMA SAYYAPARAJU, MD MUSABBIR ADNAN, SHERIF AMER, and GARRETT S. ROSE, University of Tennessee, Knoxville, USA

In the past decade, complementary metal oxide semiconductor-memristor hybrid neuromorphic systems have gained importance owing to the advantages of memristors such as nano-scale size, non-volatility, and low-power operation. However, they are often accompanied by non-ideal properties that can impact the system's performance. This article presents device-aware circuit design to mitigate such effects. A bi-memristor synapse with a robust spike-timing-dependent plasticity (STDP) is designed. A mixed-mode neuron is presented whose accumulation rate is tunable on-chip and can be used with a variety of memristors without needing a re-design. The proposed designs are employed together in an example pattern recognition system. A scalable winner-takes-all circuit is presented for the output stage. A pattern recognition task based on a simple STDP-based learning is demonstrated such that the recognition rate is directly dependent on the learnt weights. Device-level issues such as switching speed/threshold asymmetry, limited switching resolution, endurance, and varying resistance range (across devices) are shown to adversely affect learning at the system level and it is demonstrated that the proposed circuits can mitigate them. Last, the area and energy costs of the proposed designs are evaluated and compared against other implementations in the literature.

CCS Concepts: • **Hardware** → **Emerging technologies**; **Neural systems**; *Robustness*; Very large scale integration design; Analog and mixed-signal circuits;

Additional Key Words and Phrases: Memristor, synapse, neuron, learning, asymmetry

ACM Reference format:
Sagarvarma Sayyaparaju, Md Musabbir Adnan, Sherif Amer, and Garrett S. Rose. 2020. Device-aware Circuit Design for Robust Memristive Neuromorphic Systems with STDP-based Learning. *J. Emerg. Technol. Comput. Syst.* 16, 3, Article 28 (May 2020), 25 pages.
https://doi.org/10.1145/3380969

Preliminary portions of this article appeared in the *Proceedings of the 19th International Symposium on Quality Electronic Design (ISQED'18)* [43] and in the *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'18)* [44].
This material is based on research sponsored by Air Force Research Laboratory under agreement numbered FA8750-19-1-0025. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.
Authors' address: S. Sayyaparaju, Md M. Adnan, S. Amer, and G. S. Rose, University of Tennessee, Knoxville, 1520 Middle Drive, Knoxville, Tennessee, USA, 37996; emails: {ssayyapa, madnan, samer1}@vols.utk.edu, garose@utk.edu.

1550-4832/2020/05-ART28 $15.00
https://doi.org/10.1145/3380969

1 INTRODUCTION

The development of computing machines that mimic the cognitive actions of the human brain for data processing has attracted researchers' attention for many years [9]. With rapidly increasing volume of data, conventional computing based on von Neumann architecture faces performance hurdles due to, among other things, data latency and high power consumption [22]. Further, to perform cognitive tasks, software implementations of neural networks are resource and power intensive [37]. Hence, hardware neural networks that resemble biological systems by employing neurons that propagate information in the form of spikes through electrical synapses have received increased attention.

Synapses have been built using resistors [15], capacitors [33], floating gate transistors [41], and Static Random Access Memories (SRAMs) [32]. However, they do not provide a way to realize a non-volatile analog weight amenable to efficient programming [1]. Nearly a decade ago, a two-terminal non-volatile resistive switching device known as the memristor was demonstrated in Reference [49] and was shown to be suitable as a synapse implementing the bio-inspired spike-timing-dependent plasticity (STDP) rule for weight updates [24, 48]. This device was first hypothesized by Chua [11] as the missing fourth fundamental circuit element, whose resistance can be modulated by applying a voltage flux. Its nanoscale size, non-volatility, and low-power analog programmability and readability has spurred widespread research interest in memristor-based neuromorphic systems.

In the recent past, CMOS-memristor hybrid systems have been proposed and experimentally demonstrated for representative neuromorphic applications such as pattern recognition [2, 10, 28, 39, 46, 53]. Numerous other works have shown in simulation the design and operation of memristor-based neural networks with on-chip learning [8, 12, 13, 16, 29, 34–37, 40, 47, 54, 55]. However, most of the proposed implementations either have large circuit area overhead for synaptic functionality and its on-chip plasticity or they employ such spike shapes at the neuron's output and feedback, that perform only a single step potentiation/depression (not true STDP), adding extra area and potentially wasting power for long input pulses as mentioned in Reference [40]. Moreover, largely overlooked device-level issues such as switching speed/threshold asymmetry, limited switching resolution, endurance and varying resistance (across device types) can hamper the learning and functioning of neural systems and need to be accounted at the circuit design level.

Addressing the aforementioned concerns, this article presents a device-aware circuit design approach for synapses and neurons. A bi-memristor synapse with a (device-aware) robust STDP scheme is designed such that weight updates can be precisely controlled. Also, a generic mixed-mode neuron is proposed that can be used with a wide variety of memristors, thus avoiding custom re-design for each new device type. The robustness and reliability of these designs is demonstrated by employing them together in an example pattern recognition system. The following section provides a review of the prior work for the design of synapses and neurons in this context and summarizes the contributions of this work.

2 RELATED WORK

Memristor-based synapses and their plasticity have been implemented using a variety of techniques. Authors of Reference [24] have proposed an STDP scheme by applying pulses whose width is an exponential function of the relative timing of the pre- and post-neuron spikes. Similarly, in Reference [48] a time division multiplexing of pulses was proposed. However, both of these approaches need additional circuits to determine the relative timing of neurons' spikes and to generate width modulated pulses. Another STDP scheme was proposed in Reference [36] that has circuits that take into account the pre- and post-neuron spike timing when performing a weight update. This method also has the same issues as above. In Reference [37], STDP-based online

learning was demonstrated using a 1T1R synapse. However, the gating transistor per synapse nullifies the density advantage of the nanoscale memristor device.

In References [12, 16, 26, 54], a single memristor-based synapse is used with neuron spikes that are dissimilar at its output and feedback and their overlap leads to a weight change. However, these extra spike shapes need additional circuits for their generation. In Reference [19], spikes with discrete levels of voltages were used. However, these techniques could only implement an excitatory (positive weight) synapse.

Analog spikes were also adopted in Reference [45] and the influence of spike shape on the observed STDP character is shown. Inhibitory synapses were also shown with inverted spikes. In Reference [27], the neuron was configured as either inhibitory or excitatory using a second generation current conveyor circuit. In these schemes, the sign of synaptic weight is a neuron artifact and is not embedded within the synapse. In References [1, 25], a bridge-like memristor configuration was proposed that could represent both positive and negative weights. However, this scheme requires an additional differential amplifier to convert the synaptic state information of the bridge into a corresponding current.

For a simple implementation of non-volatile positive and negative synaptic weights, the use of two memristors per synapse has been proposed. Authors of Reference [46] have used two memristors, each of which drive opposite signals during a read operation. However, only potentiation has been applied to the devices for weight updates, requiring an extra "sleep cycle" during its operation where the devices are reset and the synapse is rewritten. In Reference [17], two memristors were used per synapse to obtain positive and negative weights, but STDP was not the focus and complex circuits were required for weight update. A twin-memristor synapse was proposed in Reference [6] using digital spikes for read and weight updates. However, this scheme needs an additional "control block" per synapse that takes-in spike information from the pre- and post-neuron. Moreover, these schemes have not taken into account the limitations of contemporary devices such as asymmetry of switching speed/threshold, limited resolution and endurance [42]. These non-ideal device properties can adversely affect the system's functioning and require meticulous circuit level attention.

The other key component of neuromorphic systems is the neuron. Although many models for neuron behavior have been proposed, a review of which can be found in Reference [21], most of the (spiking) systems mentioned above utilize an analog integrate and fire model for the neuron [29]. This neuron relies on the integration of incoming current and accumulation of the resulting charge. The accumulated voltage is then compared with a threshold to determine the spike condition. Hence, the rate of charge accumulation (and thereby the neuron spiking probability) is dependent on the capacitor used for integration. This implies that the neuron needs to be custom designed to suit the specific memristor used as the synapse and for the specific application. Re-design of a neuron for a new memristor type not only entails the designer's time and effort but also huge re-fabrication costs for the new lithographic masks needed during chip fabrication. Given that memristor devices can be of varied types, the cost of fabrication will go very high if neuron designs in the Front-End-of-Line (FEOL) need to be changed every time a new memristor device in the Back-End-of-Line (BEOL) needs to be tested and integrated with silicon neurons on a chip. This will have an adverse effect on the time-to-market of this technology.

Overcoming the aforementioned issues, this article presents a device-aware circuit design methodology for spiking synapses and neurons. Device-aware robust circuit design is shown, that can adapt to a variety of (realistic) constraints prevalent in contemporary memristor devices. The specific contributions of this article are as follows:

- A bi-memristor synapse is presented that can implement both positive and negative weights without any additional local circuitry at the synapse.

- An STDP mechanism for precise weight updates is achieved based on a spike that is discretized both in voltage and time, where both of them can be controlled on-chip.
- The STDP behavior and weight change magnitude of the synapse is shown to be adjustable and its resilience to the asymmetry of device switching is demonstrated.
- A generic mixed-mode neuron is proposed whose accumulation rate is tunable on-chip, thus eliminating the need for a neuron re-design with a change in the memristor type used.
- The proposed synapse and neuron are employed together in an example pattern recognition system with STDP-based learning. A simple and scaling-friendly winner-takes-all (WTA) scheme is proposed at the output layer of this system.
- It is shown that the detrimental impact on STDP and hence on the system performance due to device issues such as switching asymmetry, limited resolution, endurance, and memristor change can be remedied by virtue of using the proposed circuits.

3 MEMRISTOR MODEL

Memristor models proposed in the literature can be broadly classified into two categories: physics-inspired models and behavioral models. While physics-inspired models are often preferred, the lack of complete understanding of the physical dynamics of memristive switching has promoted the development of behavioral models that are sufficiently accurate, converge well in circuit simulation and amenable to parameter extraction. Most behavioral models start by defining an internal state variable controlled by the applied electrical excitation (voltage or current). Another relationship is then used to link the state variable with the device's resistance. This modeling paradigm may not be readily suitable for parameter extraction as the state variable is not a measurable parameter. A model based on the instantaneous resistance as the state variable was proposed in Reference [3] and has been used in this work. Adopting the instantaneous resistance as the state variable has an advantage that it can be easily extracted from I-V sweeps. This model is described as follows:

$$\frac{dM}{dt} = \begin{cases} -C_{LRS}\left(\frac{V(t)-V_{tp}}{V_{tp}}\right)^{P_{LRS}} f_{LRS}(M(t)), & V(t) > V_{tp}, \\ C_{HRS}\left(\frac{V(t)-V_{tn}}{V_{tn}}\right)^{P_{HRS}} f_{HRS}(M(t)), & V(t) < V_{tn}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where C and p are the speed and non-linearity parameters, respectively. f_{HRS} and f_{LRS} capture the resistance plateauing near the edges (commonly referred to as window functions). Equation (2) presents the window function used here that can be easily fitted to measurable parameters:

$$f(M(t)) = \begin{cases} \frac{1}{1+e^{\frac{M(t)-\theta_{HRS}HRS}{\beta_{HRS}\Delta r}}}, V(t) < V_{tn}, \\ \frac{1}{1+e^{\frac{\theta_{LRS}LRS-M(t)}{\beta_{LRS}\Delta r}}}, V(t) > V_{tp}. \end{cases} \tag{2}$$

Here, $\Delta r = HRS - LRS$ and θ and β are fitting parameters of the window function that determine the onset of the plateauing and the slope of the transition therein, respectively. For simulations in this article, the following parameters have been used: $HRS = 12K\Omega, LRS = 2.5K\Omega, V_{tp} = 0.6V$, $V_{tn} = -0.6V$, $\theta_{HRS} = 0.85$, $\theta_{LRS} = 1.6$, $\beta_{HRS} = \beta_{LRS} = 0.07$, $C_{HRS} = C_{LRS} = 9.5 \times 10^9$, $P_{HRS} = P_{LRS} = 2$. This is based on the device presented in Reference [50] that was experimentally demonstrated to have analog switching characteristics with intermediate resistance states.

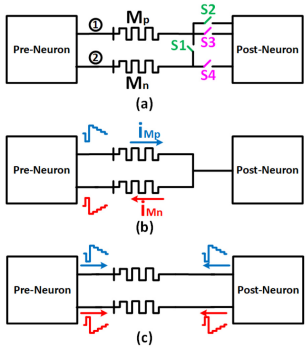

Fig. 1. (a) The bi-memristor synapse with its connections to the pre- and post-neuron. (b) The synapse configuration when the pre-neuron spikes and the post-neuron is accumulating. (c) The synapse after the post-neuron also spikes.

4 BI-MEMRISTOR SYNAPSE

4.1 Synapse Structure and Operation

The bi-memristor synapse is composed of two memristors, named M_p and M_n, connecting a pre-neuron and a post-neuron as illustrated in Figure 1(a). Note that although a similar synapse has been shown in Reference [6], it requires a "control block" per synapse, that takes-in spike information from the pre- and post-neuron. This implementation uses additional circuitry local to the synapse for weight update. The bi-memristor synapse presented here eliminates such additional local circuits and its operation can be divided into two phases: *accumulation* and *learning*. During the *accumulation* phase, as the pre-neuron spikes, the switches S1 and S2 of the post-neuron are closed, giving the configuration as shown in Figure 1(b). The pre-neuron applies spikes with opposite voltage polarity on nodes 1 and 2 as shown, whereas the other node is grounded by the post-neuron. This leads to opposite current flow in M_p and M_n, the sum of which flows into the post-neuron, and is given as

$$i = i_{M_p} - i_{M_n} = (G_p - G_n)V_{spike}. \tag{3}$$

Therefore, the effective weight of the synapse, designated by its effective conductance is given by

$$G_{eff} = G_p - G_n = \frac{1}{M_p} - \frac{1}{M_n}. \tag{4}$$

This incoming current is then "accumulated" in the post-neuron and a spike is generated once the accumulation crosses the threshold. When the post-neuron spikes, switches S1 and S2 are opened while S3 and S4 are closed, leading to the configuration shown in Figure 1(c). The post-neuron, while propagating spikes onward to the next synapse, also feeds them back to its input synapse(s). When the post-neuron spikes, the pre-neuron could be in the middle of its spiking event (they operate independently). Thereby, the memristors are biased with voltages from the spikes of both neurons and the resulting *learning* (weight change) depends on the relative timing of these spikes.

The relation between the relative timing of the neurons' spikes and the resultant voltage difference across the memristors is illustrated in Figure 2. As seen in Figure 2(a), when the post-neuron spikes after the pre-neuron, a net positive voltage across M_p crosses the threshold. Also, since the spikes associated with M_n are inverted in polarity, a net negative voltage greater than the threshold is applied across M_n. Hence, M_p decreases whereas M_n increases leading to a net increase in

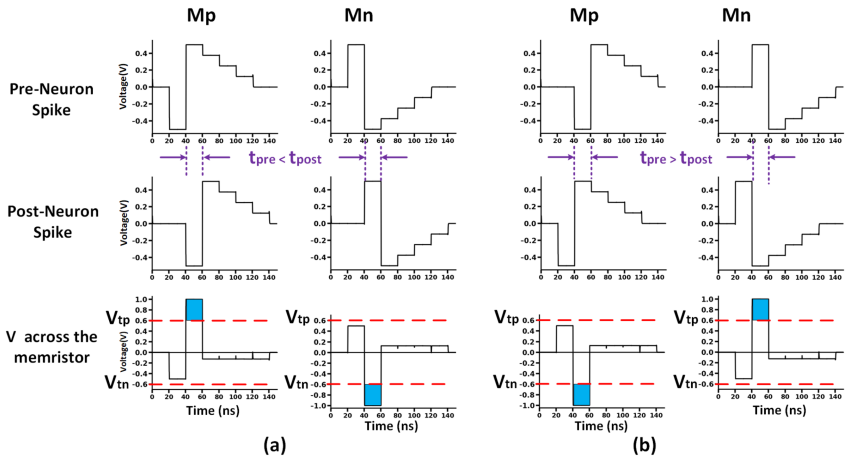

Fig. 2. The temporal occurrence of the pre- and post-neurons' spikes and the resulting voltage drop across M_p and M_n for (a) potentiation and (b) depression obtained from circuit simulations in Cadence Spectre.

G_{eff} (potentiation) as per Equation (4). The new conductance G'_{eff} can be given as

$$\begin{aligned} G'_{eff} &= \frac{1}{M_p - \Delta M} - \frac{1}{M_n + \Delta M} \\ &= \frac{1}{M_p\left(1 - \frac{\Delta M}{M_p}\right)} - \frac{1}{M_n\left(1 + \frac{\Delta M}{M_n}\right)} \\ &= \frac{1}{M_p}\left[1 + \frac{\Delta M}{M_p} + \left(\frac{\Delta M}{M_p}\right)^2 + \cdots\right] - \frac{1}{M_n}\left[1 - \frac{\Delta M}{M_n} + \left(\frac{\Delta M}{M_n}\right)^2 - \cdots\right] \\ &= \frac{1}{M_p} - \frac{1}{M_n} + \Delta M\left(\frac{1}{M_p^2} + \frac{1}{M_n^2}\right) + \Delta M^2\left(\frac{1}{M_p^3} - \frac{1}{M_n^3}\right) + \cdots \\ &= G_{eff} + \Delta M\,(G_p^2 + G_n^2) + \Delta M^2\,(G_p^3 - G_n^3) + \cdots . \end{aligned} \tag{5}$$

Therefore, the net change in the conductance after potentiation is given as

$$\begin{aligned} \Delta G &= G'_{eff} - G_{eff} \\ &= \Delta M\,(G_p^2 + G_n^2) + \Delta M^2\,(G_p^3 - G_n^3) + \cdots . \end{aligned} \tag{6}$$

Similarly, when the post-neuron spikes before the pre-neuron (a depression condition), M_p and M_n experience a net negative and positive voltage greater than the threshold, respectively, as illustrated in Figure 2(b). Hence, M_p increases and M_n decreases in this case, leading to a net decrease in G_{eff}. The corresponding change in the conductance is given by

$$\Delta G = -[\Delta M(G_p^2 + G_n^2) - \Delta M^2(G_p^3 - G_n^3) + \cdots]. \tag{7}$$

It can be seen from Equations (6) and (7) that the ΔG expression resembles the series expansion of an exponential function, where ΔM is the independent variable. As noted in Section 3, the change in memristance is of the format $\Delta M = kV^p$, where V is the effective voltage applied across the memristor and k is a constant dependent on the device's switching mechanism. Hence, the change in the conductance can also be represented as

$$\Delta G = k_1 V^p + k_2 V^{2p} + k_3 V^{3p} + \cdots . \tag{8}$$

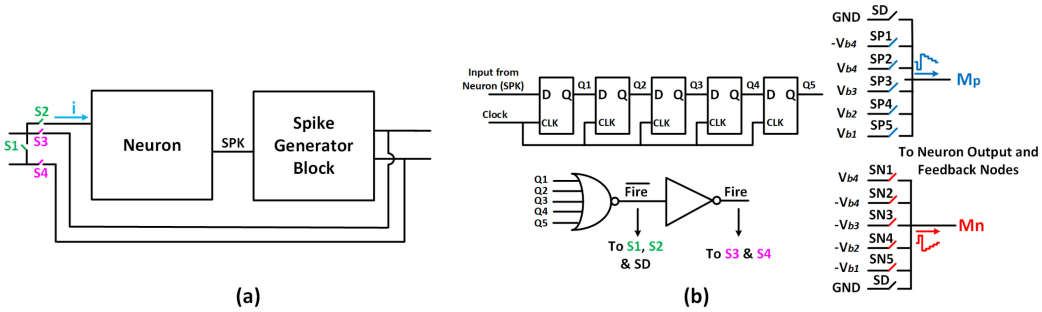

Fig. 3. (a) The neuron with the spike generation block and the control switches (b) Spike generation block.

It is evident from Equation (8) that ΔG is an exponential-like function of the voltage across the devices. Moreover, the neuron spikes in Figure 2 are shaped such that the voltage changes linearly after each clock period. Hence, the net voltage V across devices also changes linearly with respect to $\Delta T = t_{post} - t_{pre}$, the temporal difference between the pre- and post-neurons' spikes. Therefore, ΔG is exponentially dependent on ΔT, thus implementing an exponential STDP behavior [29].

4.2 Spike Generation Block

The block diagram of the neuron integrated with the spike generator block along with its control switches is shown in Figure 3(a). As explained in Section 4.1, during the accumulation mode, switches S1 and S2 close (S3 and S4 open), thus summing the input currents from M_p and M_n. The total current i flows into the neuron and is "*accumulated*" therein. When the accumulation crosses the threshold, the neuron spikes and enters its *refractory period*. During this period, the neuron propagates its spikes onward to its output synapses and also sends the same spikes as a feedback to its input synapses by opening S1 and S2 and closing S3 and S4.

The spike generator block is shown in Figure 3(b). It consists of two groups of switches, one each to control spikes for M_p and M_n. The switches are controlled by the nodes Q1–Q5 such that each node controls a particular switch in each group. As the spike from the neuron is input to this circuit, switches SP1 and SN1 are closed first. At each clock edge, the spike propagates through the nodes Q1–Q5 and therefore one switch among SP1–SP5 and SN1–SN5 is closed sequentially, while the rest of the switches remain open. For each switch that is closed, the output spike receives the corresponding voltage as shown in the figure. Note that the voltages here are linearly graded and can be produced using an on-chip voltage divider. Also, when there is no spike, the default switches SD are closed, biasing the memristors at half-rail (GND in this case) to avoid sneak paths in a crossbar configuration.

4.3 STDP Behavior of the Bi-Memristor Synapse

4.3.1 STDP Characteristics. To study the STDP behavior of the bi-memristor synapse, simulations have been carried out using Cadence Virtuoso with Spectre as the simulator. A 65-nm design kit has been used to implement the CMOS portions, while the memristor model described in Section 3 was written in Verilog-A and used with the device parameters mentioned therein.

To simulate the STDP behavior, the setup shown in Figure 4(a) has been used. It consists of two synapses S1 and S2 connected between neuron pairs N1–N3 and N2–N3, respectively. S2 is configured to have a high positive weight such that when N2 spikes, the accumulation in N3 is enough to make it spike. This way, the timing of the N3 spike is fixed and N1 spike is controlled to perform the STDP in synapse S1. S1 is initially configured to have a weight of zero, with

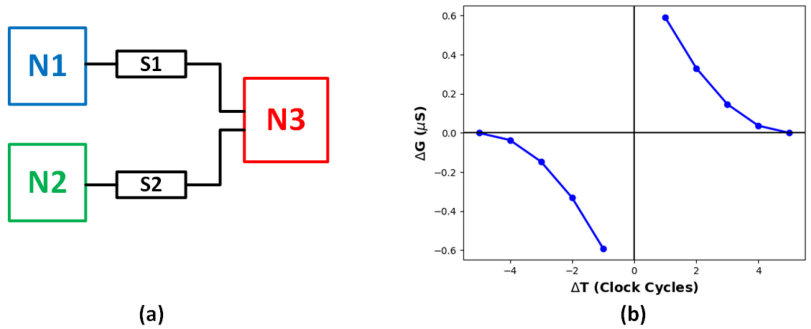

Fig. 4. (a) The simulation setup to study the STDP behavior of the synapse. (b) The STDP observed in the proposed synapse.

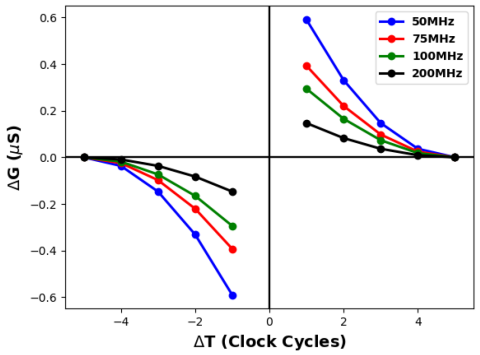

Fig. 5. The dependence of STDP characteristics on clock frequency.

$M_p = M_n$. The resultant STDP is shown in Figure 4(b). Note that this simulation has been performed at a clock frequency of 50 MHz.

4.3.2 Effect of Clock Frequency. In Figure 2, the neuron spike's temporal length is a function of the clock frequency. The longer the clock period, the longer each voltage level is sustained and hence the larger the ΔM during weight change. Hence, by controlling the clock frequency, we can fine tune the weight increments/decrements and hence the STDP character. Figure 5 shows STDP dependence on the clock frequency. As expected, a lower clock frequency (higher clock period) leads to higher ΔG and a steeper STDP curve, whereas the opposite happens with higher frequency. Moreover, for higher frequencies, when ΔM becomes small, the higher-order terms in Equation (6) can be neglected and hence it can be reduced to $\Delta G \approx (G_p^2 + G_n^2)\Delta M = k\Delta M$, suggesting a linear STDP behavior. This tendency can be observed in Figure 5 at higher frequencies.

4.3.3 Effect of Device Switching Asymmetry. Due to the fundamentally distinct switching dynamics in the memristor during resistance increase and decrease, contemporary devices often have faster switching in one direction compared to the other. This switching speed asymmetry can be as high as two orders of magnitude [4, 18] and can hamper the learning of a synapse, since it involves switching in both directions. This implies that in case of a huge asymmetry, the device that is switching in the direction of faster speed will experience a larger change and hence will dominate the overall change in weight. These large changes can affect the resolution of ΔG during

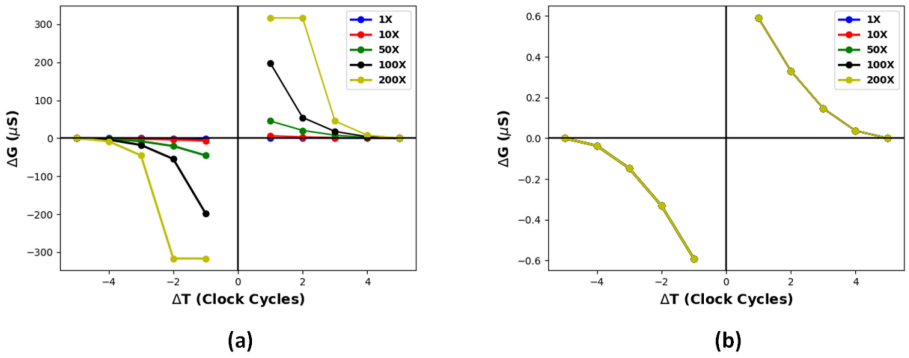

Fig. 6. (a) STDP behavior in the presence of switching speed asymmetry, where "X" represents C_{LRS}/C_{HRS}. (b) STDP behavior rectification after duty cycle modulation.

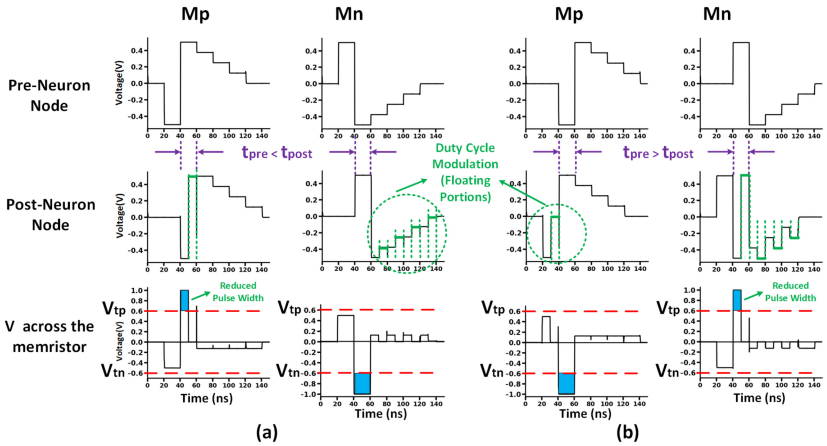

Fig. 7. Duty cycle modulation of neuron spikes to alleviate switching asymmetry effects for (a) Potentiation (b) Depression. Note that the pre neuron spikes here are unaltered and due to the duty cycle modulation, (in a given clock period) portions of the post neuron spike are "cut off," leaving that node floating (hence it follows the pre-neuron node for that part). Therefore, for this portion, the net bias across the synapse is zero.

learning (potentially impacting the granularity of the synaptic weights learnt in a neuromorphic system) and can also worsen the STDP behavior of the synapse as shown in Figure 6(a).

The adverse effects of switching speed asymmetry can be remedied by the virtue of the discretized spike shape proposed here. By reducing the flux applied to the faster switching direction, we can equalize the resistance change in both directions. To implement this, duty cycle modulation has been adopted. By controlling the duty cycle of the spike shape, the flux applied is controlled. This is illustrated in Figure 7 for the case where resistance decrease is much faster than resistance increase ($C_{LRS} > C_{HRS}$). Here, the neuron's output spikes responsible for accumulation are left unchanged, whereas only the feedback spikes are modulated. The new spike shapes are as shown in Figure 7. The reason for these shapes is as follows: during weight change, the memristance decreases when a net positive voltage is applied across the device. This occurs due to a positive pulse from the pre-neuron and a negative one from the post-neuron. Hence, the negative voltage portion of the post-neuron feedback is curtailed in time as shown in Figure 7. To produce this modulation, switches S3 and S4 need to be controlled such that they are opened whenever the

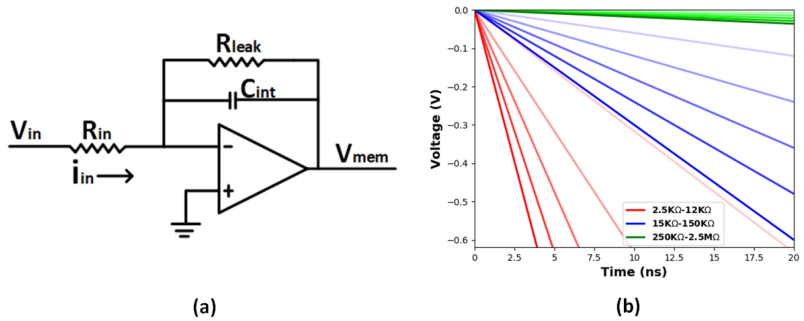

Fig. 8. (a) The integrator of an analog integrate-and-fire neuron. (b) The accumulation of voltage V_{mem} in the integrator for three device types. Here, multiple lines for each device type indicate different input synapse weights, with the lighter shade representing low weight and the darker shade standing for high weight.

Table 1. The Three Devices Considered in This Work with a Wide Range of Resistances

	Mem_1[50]	Mem_2[30]	Mem_3[23]
LRS	2.5 kΩ	15 kΩ	250 kΩ
HRS	12 kΩ	150 kΩ	2.5 MΩ

voltages are to be shortened in time. For this, the corresponding control of these switches ("*Fire*" signal in Figure 3(b)) is logically ANDed with a duty cycle modulated clock. Note that this duty cycle modulated clock can be generated from the global clock of the system using a programmable delay circuit as in Reference [31]. A "delayed" version of the global clock (say "d-CLK") can be produced with this circuit and the logical operation (d-CLK⊕CLK)·CLK yields a duty cycle modulated clock. Hence, owing to the discretized spikes adopted here, this system can cope with and rectify the switching rate asymmetry effect prevalent in memristor devices.

5 MIXED-MODE NEURON

Conventional analog integrate-and-fire neurons operate on the incoming synaptic current by integrating it using a capacitor-based integrator and store it as an 'accumulated' voltage as shown in Figure 8(a). The accumulated voltage, expressed as $V_{mem} = -\frac{1}{R_{in}C_{int}}\int_0^t V_{in}dt$ is a function of the capacitor (C_{int}) used; the larger it is, the slower the accumulation. This implies that to obtain a detectable accumulation, the value of C_{int} must be meticulously designed for the synapse (R_{in}) under consideration. However, a wide variety of memristor devices have been proposed in the literature with a range of resistance values. Hence, when a new memristor device is to be used, the neuron is likely to fail to accumulate voltages to reasonable values. Here, three such devices have been considered with different resistance values as shown in Table 1. Note that these devices have been demonstrated experimentally to have analog switching with intermediate resistance states, making them suitable for use as synapses.

The neuron's response when designed and simulated for the device Mem_2 is shown in Figure 8(b). Five different synaptic weights (set as G_{max}/x, where x=1-5 and $G_{max} = \frac{1}{LRS} - \frac{1}{HRS}$) were used at the input and it is seen here that the accumulation rate changes with weight; higher weights (darker shades in Figure 8(b)) leading to faster accumulation. The same neuron was also used with the other two synapses. For the case of device Mem_1, the resistance is smaller than that of device Mem_2, hence producing more input current. Since the designed capacitor for Mem_2 is too

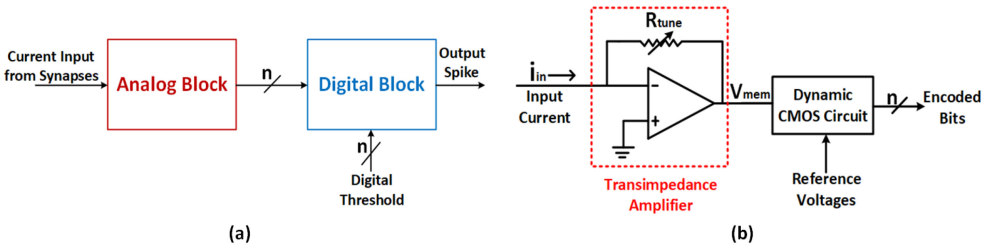

Fig. 9. (a) Block diagram of the proposed mixed-mode neuron. (b) Analog block of the mixed-mode neuron responsible for encoding the input current as digital bits.

small for Mem_1, the accumulation rate is too fast, leading to very high V_{mem} values to work with. Similarly, when Mem_3 was employed with the neuron, the resistance of Mem_3 being greater than that of device Mem_2, it produced smaller currents. The capacitance of the neuron proved too large for this current, leading to too slow accumulation and thereby too small voltages to be detected and differentiated.

The analysis above indicates that the neuron needs to be re-designed for each new type of device that is to be used with it. Since a variety of contemporary memristors are available, each with its own set of advantages, developing different systems to tap their specific desirable properties entails re-designing and fabricating a system for each new device type, proving costly in terms of design time, effort and re-fabrication costs (for the new FEOL masks). To eliminate this issue, a generic mixed-mode neuron is proposed here, whose accumulation rate is tunable on-chip.

The block diagram of the proposed neuron is shown in Figure 9(a). The incoming current from the synapses is assigned a digital value by the analog block ($n = 3$ here), based on its magnitude. This digital value is then accumulated in the digital block and stored therein. A digital threshold value is used to compare the stored value with it and produce a spike when the threshold is exceeded.

The details of the analog block are shown in Figure 9(b). It consists of a transimpedance amplifier, that converts the input current i_{in} into a voltage V_{mem} (note that V_{mem} is negative due an inverting configuration of the operational amplifier). This voltage is then input to a dynamic CMOS block that assigns a digital value to it based on the result of comparison with some reference voltages. The higher the incoming current magnitude, the higher the voltage magnitude V_{mem} and hence the higher the digital value assigned. Since V_{mem} is dependent on the tunable resistor R_{tune}, it can be implemented with transistors operating in the linear region such that R_{tune} can be tuned on-chip (with the gate voltage) to obtain the same V_{mem} for a given abstract synaptic weight across all device types. This on-chip tunability enables the neuron to adapt to various devices and have the same accumulation for all of them, thus eliminating the need for re-design.

The dynamic CMOS block (Figure 10) consists of pairs of transistors, each of which determines if the input current is within a certain magnitude range. The pairs consist of a pre-charge and a discharge transistor, which are responsible for charging-up and discharging the *evaluation node*, respectively. During the positive half of the clock period, the pre-charge transistor is ON and charges the evaluation node. During the negative half of the clock period, the pre-charge transistor is OFF. The discharge transistor has V_{mem} as the gate voltage and a reference voltage at its source. Thus, V_{mem} determines the state of this transistor. If the V_{mem} is sufficient enough to switch ON this transistor, then it discharges the evaluation node. The V_{ref} values (which can be written as $V_{mem} - V_{th}$) are graded such that $V_{ref1} < V_{ref2} < \cdots < V_{ref7}$. Note that these values must be separated enough to account for the expected resistive drops and noise effects in a big system design. Thus, with different magnitudes of input current, different V_{mem} values are

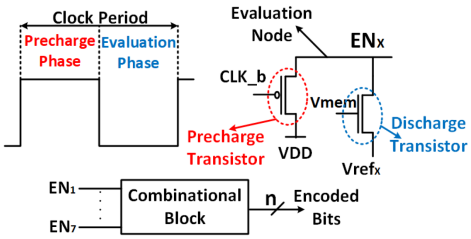

Fig. 10. The dynamic CMOS circuit to convert V_{mem} into a digital value, where $x = 1$–7. The EN_x nodes' states are encoded as shown in Table 2.

Table 2. The Truth Table for Encoding Bits Based on the Discharging of Dynamic Nodes

Nodes Discharged	Encoded Value
EN_{1-7}	000
EN_{1-6}	001
EN_{1-5}	010
EN_{1-4}	011
EN_{1-3}	100
EN_{1-2}	101
EN_1	110
None	111

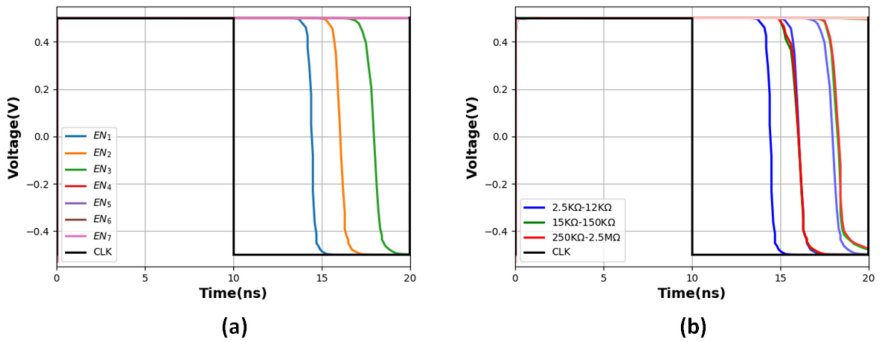

Fig. 11. (a) Simulation result for the evaluation nodes in the dynamic CMOS block showing the case for a "100" accumulation, where EN_{1-3} are discharged. (b) Similar behavior observed in the nodes for the three different device types after on-chip tuning of accumulation.

generated, such that with increasing current, V_{mem} gets more negative, making it difficult for the NMOS transistors to switch ON and hence, less number of evaluation nodes are discharged at the end of evaluation phase in a clock period. Therefore, for a high current, V_{mem} is highly negative and none of the nodes discharge, while for a very low current, V_{mem} is much less negative and all of the nodes discharge. Note that the capacitance of the node EN_x comprises the drain-body (C_{db}) and gate-drain capacitances (C_{gd}) of the NMOS and PMOS devices therein. Hence these devices need to be sized appropriately such that these capacitances charge/discharge reasonably within the clock period used. Based on the number of discharged nodes, a logic block generates a 3-bit value, which is the assigned digital value to the input current, as shown in Table 2. Figure 11(a) shows the (simulated) behavior of the evaluation nodes for a case where the accumulation is "100." For this case, nodes EN_{1-3} are discharged fully before the end of the evaluation phase. In Figure 11(b), when the synaptic device type is changed, R_{tune} is tuned on-chip to obtain the same V_{mem} and hence the same discharge behavior is seen for these nodes (for the same input weight). This demonstrates that the neuron can be tuned on-chip to act the same for various device types without any change of design.

The digital value generated by the analog block is input to a digital block that contains an adder to add the incoming value to the already stored value therein and accumulate it on the register block as shown in Figure 12. To generate a spike, the accumulated digital value is compared with an input digital threshold. Additionally, a spike is also generated when there is an overflow in the

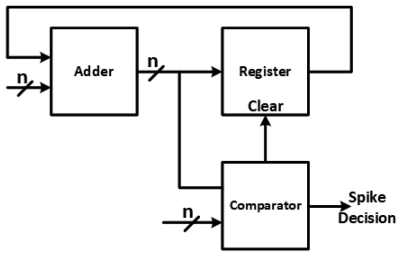

Fig. 12. Digital half of the neuron for digital mode accumulation and comparison for spike generation.

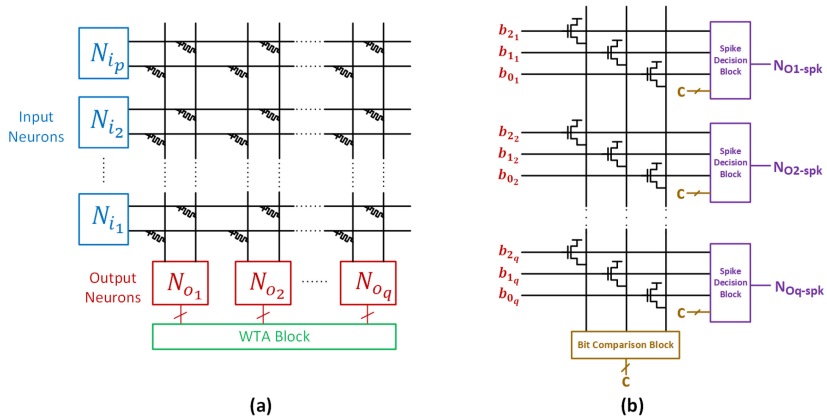

Fig. 13. (a) Block diagram of a neuromorphic crossbar system implemented with the proposed bi-memristor synapse. (b) Circuit-level diagram of the WTA system implementing bit comparison for each bit column and its use for spike decision at each neuron.

adder block. The digital comparator block determines both the comparison result and the overflow condition, and generates a spike decision. Whenever the spike condition from the comparator is true, the register block is RESET, implementing a refractory period for the neuron.

6 PATTERN RECOGNITION SYSTEM

In this section, the proposed bi-memristor synapse and the mixed-mode neuron are employed together to build and demonstrate a system for a representative pattern recognition task. The system structure and its operation using the proposed circuits is described in the following subsections.

6.1 System Structure and Operation Scheme

The block diagram of the system is shown in Figure 13(a). It consists of two layers of neurons fully interconnected by synapses in a crossbar fashion. The layer of p input neurons $N_{i_{1\ldots p}}$ apply a spike pattern to the system that is representative of the input pattern. This spike pattern generates currents through the connected synapses and causes accumulation in the q output neurons $N_{o_{1\ldots q}}$ as shown. Based on the accumulation therein, the winner-takes-all (WTA) block arbitrates between the output neurons, choosing a "winner neuron" (the one with the most accumulation). The output neurons here are different in the only aspect that their accumulation is passed to the WTA block.

The implementation of the WTA block is shown in Figure 13(b). This circuit operates as follows: to determine the winner neuron, the accumulated values in the output neurons are considered bit-by-bit, with highest priority to the most significant bit (MSB). For example, if the accumulated

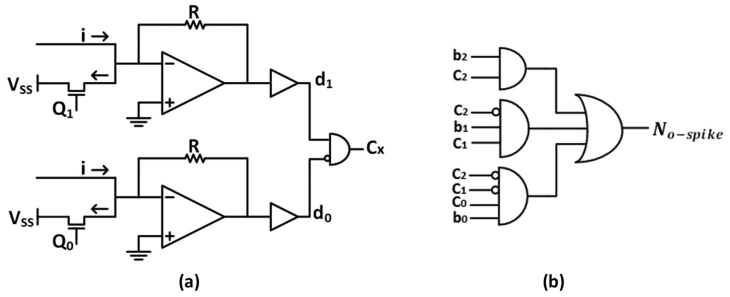

Fig. 14. (a) The bit comparison circuit at each bit line to ascertain the number of neurons with a 1 for that bit. (b) The spike decision block at each neuron to determine if it is the winner among all the output neurons.

value in a neuron N_{o_m} is $b_{2_m} b_{1_m} b_{0_m}$, where b_{2_m} is the MSB, then it is the winner if $b_{2_m} = 1$ and no other neuron's MSB is a 1, that is to say $\sum_{r=1, r\neq m}^{r=q} b_{2_r} = 0$. If any of the remaining bits $b_{2_{\{1..q\}-m}}$ are 1, or if all of the bits $b_{2_{\{1..q\}}}$ are 0, then the next bit b_{1_m} is compared with $b_{1_{\{1..q\}-m}}$ in a similar fashion. Similarly, if $b_{1_m} = 1$ and if any of $b_{1_{\{1..q\}-m}}$ is 1 or if all of $b_{1_{\{1..q\}}}$ are 0, then the evaluation moves to b_{0_m}.

To facilitate this decision making, all of the bits from the neurons are used to operate switches as shown in Figure 13(b). For a given bit b_x ($x = 1, 2, 3$), the switches from all the neurons are connected such that their ON resistances are in parallel. As explained above, to decide if a given accumulation in a neuron is the highest, three conditions need to be detected for each bit: (i) all neurons have 0 for that bit, (ii) only one neuron has a 1, and (iii) more than one neuron has a 1. These conditions must be used in conjunction with each neuron's accumulated bits to decide if its value is the highest. To differentiate between these conditions, the parallel-connected switches (size 1×) for each bit drives current into a bit comparison block, shown in Figure 14(a). This block contains switches that drive opposite currents and are sized such that the resultant current is used to detect these conditions. The 1.5× sized switch Q_1 detects if there is more than one neuron with a value 1. If more than one neuron has a 1, then at least two switches (effectively 2× size or more) drive positive current into the Op Amp (making the net positive) and hence *d1* is 0, and it is 1 if one or less neurons have a value 1 for that bit. The 0.75× switch Q_0 differentiates between one or more neurons having a 1 and none of them having a 1. Similarly, *d0* is 0 if one or more neurons have a 1 and is 0 if all of them have a 0 for that bit. Thus, a combination of these two comparisons, c_x ($x = 1, 2, 3$) helps in concluding the count of neurons with a value 1. c_x is 1 if only one neuron in a given bit column has a 1, whereas it is 0 if more than one of them have a 1 or if all of them have a 0. This bit comparison for all three bits is passed on to all the neurons' spike decision blocks. This block has the combinational logic (shown in Figure 14(b)) that decides if a given neuron has the highest accumulation, as explained in the paragraph above: a given neuron has the highest accumulation if $c_2 = 1$ and its $b_2 = 1$. If $c_2 = 0$, then this evaluation moves to bit b1 and so on. As a proof of concept, the proposed WTA system was implemented in Cadence Virtuoso and an example simulation was run, whose results are tabulated in Table 3. 10 accumulated values (from 10 neurons) are provided as input to the system and as seen therein, the system appropriately chose the winning value and did not choose anyone where there were multiple inputs with highest value.

After the WTA block makes the spike decision for each neuron, the logic output N_{o-spk} in Figure 13(b) controls the spike feedback switches (S3 & S4) for each neuron. Since this logic will hold TRUE for a maximum of only one neuron, only the winning neuron can propagate a feedback spike (that aids STDP in the synapses). Also, it must be noted that since the spike propagation is gated locally at each neuron (by N_{o-spk}), the spike generating circuit can be shared among all

Table 3. The Simulation Results for the WTA Block Handling the Cases of Only One Neuron Having a Highest Accumulation and Multiple of Them Having the Same Highest Value

Neuron	Accumulation	Spike Decision
N_{o_1}	001	0
N_{o_2}	010	0
N_{o_3}	001	0
N_{o_4}	100	0
N_{o_5}	110	1
N_{o_6}	011	0
N_{o_7}	001	0
N_{o_8}	010	0
N_{o_9}	100	0
$N_{o_{10}}$	011	0

Neuron	Accumulation	Spike Decision
N_{o_1}	001	0
N_{o_2}	111	0
N_{o_3}	001	0
N_{o_4}	100	0
N_{o_5}	010	0
N_{o_6}	011	0
N_{o_7}	001	0
N_{o_8}	111	0
N_{o_9}	100	0
$N_{o_{10}}$	011	0

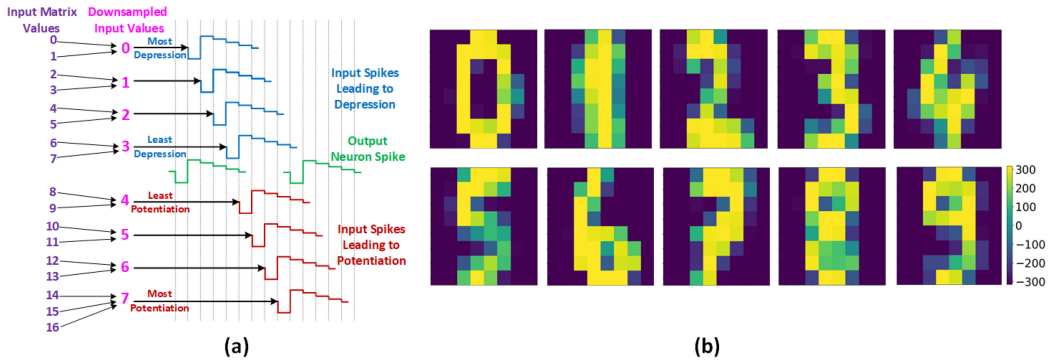

Fig. 15. (a) Input matrix downsampling and corresponding input spike temporal encoding. (b) Final synaptic weights of the system representing learnt patterns. The colorbar indicates synaptic conductances in μS.

the output neurons of the system, thus drastically reducing the overhead of the system and also helping facilitate system scaling.

6.2 Training

To simulate a pattern recognition task, the system shown in Figure 13 and its behavior as described above were modeled using Python. It operates on the handwritten digits data set from the UCI Machine Learning Repository [14]. This data set consists of input matrices of 8x8 size representing instances of digits from 0 to 9. The elements of these matrices are integers in the range 0 to 16. There are a total of 3,823 instances of digits from all the 10 classes in the training set and an independent set of 1,797 instances meant for testing. Since the input matrices here are of size 8×8 with a total of 10 classes of patterns, the system consists of 64 input neurons and 10 output neurons, each representing an input matrix element and a pattern class, respectively. The values of the input elements range from 0 to 16, and are downsampled as values from 0 to 7 such that 0–1 in the input is coded as 0, 2–3 as 1, and so on, as shown in Figure 15(a).

During training, the system is provided with input spike patterns with a temporal encoding to represent the input pattern as shown in Figure 15(a). The downsampled values of the inputs are used as delays (in units of clock periods) in triggering the corresponding input neuron's spike.

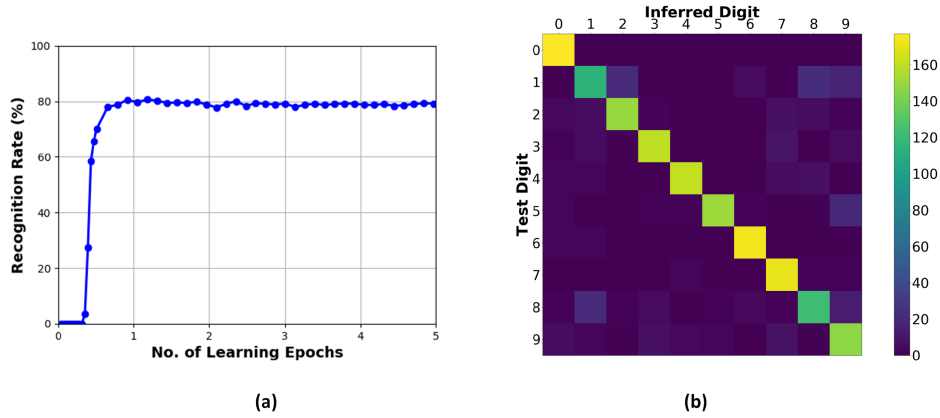

Fig. 16. (a) Recognition rate achieved on the system after several epochs of training (b) Confusion matrix for the tests performed by presenting all of the 1,797 test patterns.

The input with value 0 spikes first (say at time t_i), the one with value 1 spikes after 1 clock period ($t_i + 1$) and similarly, the input with value 7 spikes after 7 clock periods ($t_i + 7$). Also, the class labels of the input patterns are read into the simulator and are used to trigger the output neuron spike, thus facilitating STDP-based learning. For each input pattern presented, the output neuron corresponding to the input label is made to spike 1 clock period before the the input time span begins (time t_{i-1}) and also 1 clock period after it (time t_{i+1}). Since the spike waveform in this work has four different levels of positive voltages spread over a span of four clock periods, STDP can occur when two spikes are separated by up to four clock periods in time. Hence, after the first output neuron spike, the synapses connected to the inputs with values 0–3 will be depressed, with the synapse with 0-input having the highest depression. Similarly, synapses with 4–7 as the input will be potentiated by the output spike at the end of input time span. Note that the 1st spike of the output neuron has no impact on this case, since it is separated too far in time to have an overlap.

The entire 3,823 training set patterns are presented to the system for learning. The synapses in the crossbar are all initially set to a zero weight ($M_p = M_n = HRS$). As learning progresses, the weights evolve, with synapses being both potentiated and depressed. The eventual *learnt weight* of a synapse depends on the relative number of potentiations and depressions. For synapses that mostly receive inputs in the range 4–7, the net potentiation outweighs depression, hence those synapses reach a high positive weight. The opposite happens with synapses receiving inputs mostly in the range 0–3. For synapses that receive comparable number of inputs in both the ranges, the final weight saturates in between the positive and negative extremes. This behavior can be observed in Figure 15(b), where the learnt weights on all the 10 columns of the system are plotted as 8×8 patterns.

6.3 Testing

The trained system was tested by applying all of the 1,797 patterns from the testing set. During testing, the currents flowing out of each column lead to a digital value accumulation in the output-neurons. The WTA block acts as the judge to decide on the "winner." If only a single neuron has the highest accumulated value, then it is declared the winner and its label is compared with the input label to check for the test case's success. If the labels do not match, or if the highest value was attained by multiple neurons, then the case is said to have a wrong inference. These test results are shown in Figure 16(a). It is seen that the test accuracy attains its maximum after the first epoch

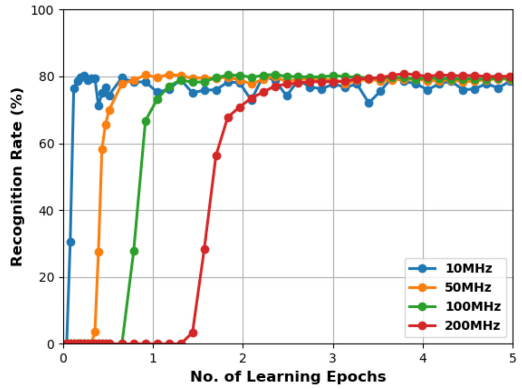

Fig. 17. The dependence of learning rate on the clock frequency used.

and it remains saturated beyond that point. The recognition rate obtained here is on par with the 83% achieved in literature [42]. Also, it must be noted that only a simple STDP-based learning on a single crossbar was used, such that the learnt weights are dependent on the efficiency of STDP. This aids in the evaluation of the robustness of the proposed designs at the system level, as demonstrated in Section 7, which is the main focus of this work. Also, the confusion matrix for these tests is shown in Figure 16(b). It can be seen here that test digit was in most of the cases inferred as the correct one (as indicated by the diagonal boxes). Among the commonly confused bits were 8 and 1, which is likely caused by the high conductance synapses in the central columns of both of these patterns as seen in Figure 15(b). Additionally, 1 was confused with 2 and 5 with 9, which can be similarly attributed to the overlap of the location of most of the high conductance synapses in these patterns.

7 PERFORMANCE ANALYSIS

In this section, we analyse the performance of the proposed designs under several practical device/circuit considerations and demonstrate their robustness against these factors.

7.1 Effect of Clock Frequency

The amount of weight change during STDP (as seen in Figure 5) in this approach is dependent on the choice of clock frequency. As can be seen, higher clock frequency leads to smaller ΔG increments, whereas lower frequency leads to larger increments. This impact of clock frequency on the weight increments and thereby on the learning curve of the pattern recognition system is shown in Figure 17.

From Figure 17, it can be seen that for higher frequencies, the learning is slower due to smaller ΔG. Hence, the learning rate can be increased by reducing the frequency. However, by decreasing the frequency disproportionately, it is seen here that the learning does not have a smooth saturation. This is due to the larger ΔG at low frequency operation that leads to poor granularity of weights.

Therefore, based on this trade-off between learning rate and weight granularity, an optimal frequency must be chosen, which in this case is 50 MHz, at which the system's learning saturates after about 1 epoch of training and remains constant thereafter. Note that this analysis highlights an important device metric: resolution of resistance states. As seen here, low resolution of resistance states yields poor control over the granularity of weights in the system. Hence, higher resolution devices with greater control over achievable intermediate resistance states are necessary. The

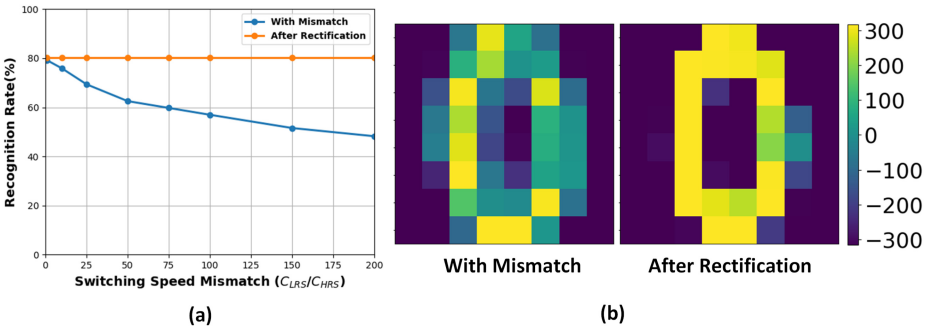

Fig. 18. The behavior of the pattern recognition system illustrated in terms of (a) recognition rate, (b) learnt pattern "0," before and after the switching speed asymmetry effect is remedied.

approach adopted in this work enhances this capability by providing the designer with the choice of frequency, with which the resolution of weight changes can be controlled and fine tuned.

7.2 Device Switching Asymmetry

7.2.1 Switching Speed Asymmetry. It was shown in Section 4.3.3 that an asymmetry in the switching speed of the memristor leads to drastic changes in weights during STDP. This condition is simulated for the pattern recognition system and shown in Figure 18(a). It is seen that as the asymmetry in the device increases, the accuracy drops. As also described in Section 4.3.3, by adopting a duty cycle modulation technique in the feedback spike, this drop in the accuracy can be rectified as shown in Figure 18(a). Additionally, Figure 18(b) illustrates an example of a (disfigured) learnt pattern in the presence of asymmetry and the rectified pattern after using the duty cycle modulation technique.

7.2.2 Switching Threshold Asymmetry. Apart from switching speed asymmetry, another important non-ideality is the switching threshold asymmetry prevalent in contemporary devices. As seen in Equation (1), ΔM is a function of the voltage overdrive $V - V_t$ across the device. Hence, when V_t in one direction is higher than the other, the overdrive across the device in that direction will be less, causing reduced switching or no switching if the overdrive is zero. This will hamper the STDP behavior of the synapse and hence the learning of the system. This effect has been simulated and shown in Figure 19(a) along with the effect of switching speed asymmetry. Here, both of theses effects are plotted separately, as well as in combination. It can be seen that when both of these effects occur in combination, the result is catastrophic, leading to a rapid failure of the system.

An advantage of using the constant, discrete voltage levels in the spike shape as shown in Figure 2 is that the designer has the option of assigning the voltage of his choice to each level. Hence, in the case where there is a reduced overdrive due to a higher V_t, it can compensated by increasing the voltage levels in the feedback spikes as shown in Figure 19(b). For this particular example, it is assumed that $\|V_{tn}\| > \|V_{tp}\|$. Hence, the net negative voltage applied across the device during weight change must be increased to compensate for the higher V_{tn}. A net negative voltage is applied (across M_n during potentiation and M_p during depression) by the overlap of a negative pulse from the pre-neuron end and a positive pulse from the post-neuron's feedback. Hence, the positive pulses of the feedback are increased in magnitude here. When both the switching speed and threshold have an asymmetry, a combination of duty cycle modulation and pulse voltage control can be employed for compensation. The result for this compensation is shown in Figure 19(a). It is seen here that by using a combination of the compensation techniques in time and voltage,

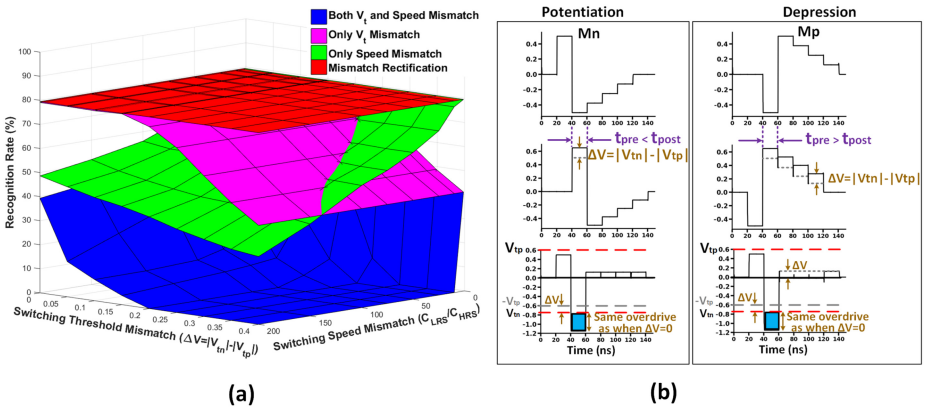

Fig. 19. (a) Recognition rate of the system in the presence of switching asymmetry effects and their rectification by the proposed methods. (b) Voltage modulation in the feedback spikes to compensate for threshold asymmetry, demonstrated through simulation in Cadence Spectre.

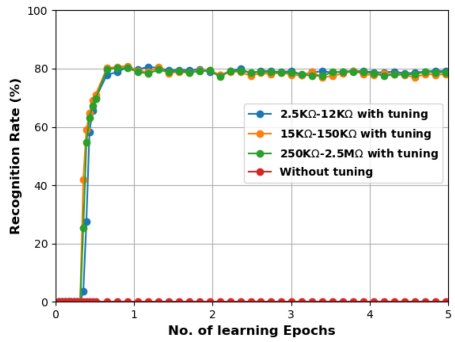

Fig. 20. The pattern classification accuracy for various devices remains the same with the on-chip tunability in the proposed neuron. Note that here "without tuning" refers to a test conducted with a "rigid" neuron designed for a specific device (2.5–12KΩ in this case) employed with other two mentioned device types.

the switching asymmetry issue can be mitigated and the original recognition rate is restored. Note that the explicit analysis and mitigation methods discussed here differentiate this work from those described in Section 2.

7.3 Effect of Change in Memristor Device

As explained in Section 5, a wide range of memristor resistance values have been demonstrated in literature, and a neuron designed for a given device is likely to fail when employed with a different device. This is demonstrated at the system level in Figure 20. Here, the proposed mixed-mode neuron was first used without the tunability feature, wherein it was designed for one particular memristor device (2.5–12 KΩ). It is seen here that when it was employed with other two devices, the classification fails. However, with the tunability feature added, the neuron is able to adapt to other device types and the classification is restored and is consistent across device types. Note that this was made possible by the transimpedance amplifier employed in the proposed neuron and the use of tunable resistors. This demonstrates that the proposed neuron is generic and can prevent the need for custom design, re-design and re-fabrication of the FEOL circuits in memristor neuromorphic systems where several device types need to be actively prototyped and experimentally tested.

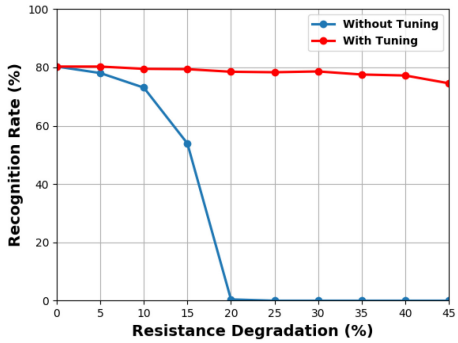

Fig. 21. On-chip tunability of the neuron applied to mitigate the effects of limited device switching endurance, leading to resistance degradation (shown here as % deviation from their initial value).

7.4 Switching Endurance

Switching endurance in memristor devices is another important reliability concern. With increasing number of switching cycles that the device experiences, the LRS and/or HRS of the device deviate from normal, with an increase in LRS and decrease of HRS being common [7, 38, 52]. This leads to a decrease in the *HRS*/*LRS* ratio, known as the switching window of the device, as illustrated in Reference [7]. This impact has been simulated at the system level by gradually increasing and decreasing the LRS and HRS values of the device, respectively. With a non-tunable neuron, Figure 21 shows that with the drifting resistances, the recognition rate also degrades. However, with the proposed tunable neuron, the accumulation rate can be adjusted as the device properties change, and hence the classification can be recovered as shown in Figure 21. Note that at 45% degradation of LRS and HRS for the 2.5–12KΩ device (where the recognition rate starts to roll down), the switching window is already below 2×. Hence, the proposed neuron helps alleviate endurance-related reliability concerns in memristive neuromorphic systems. Also, this analysis highlights another important feature of the proposed neuron: adaptability to shrinking switching windows. Figure 21 illustrates that the proposed neuron can be tuned to operate with a small switching window.

8 DISCUSSION

In this section, various implementation aspects of the designs proposed above are discussed.

8.1 Choice of Number of Bits—"n"

As seen in Figure 9, the input current to the mixed-mode neuron is assigned an "n" bit digital value, which was chosen as 3 in the work above. The impact of varying this value was simulated and is shown in Figure 22. It is seen here that the accuracy starts to plateau for $n \geq 3$ and was 80%, 84% and 84.75% for $n = 3$, $n = 4$, and $n = 5$, respectively. Since the increase in accuracy is not significant beyond n=3, it remains a good choice (to minimize area overhead). Also, it may be noted that in Reference [42], an accuracy of 83% was reported on the same dataset. Given the focus on robustness of the system proposed here, the pattern recognition accuracy obtained is on par with that in the literature.

8.2 Area Overhead and Energy Consumption

To estimate the area overhead of the proposed neuron, its physical design was performed and the layout area is shown in Table 4. It is seen here that the area of the proposed neuron is lower

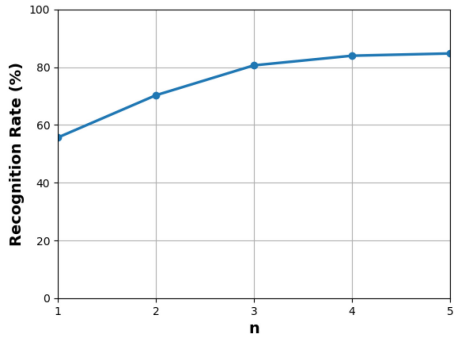

Fig. 22. The dependence of accuracy of the system on "n."

Table 4. Layout Area Comparison for Neuron Implementations

	This Work	Integrate-and-Fire Neuron [6, 44]
Layout Area (μm^2)	1,393 (65 nm)	~2,000 (65 nm)

Table 5. Comparison Table for Energy Consumption per Spike of Various Proposed Designs in Literature

	Energy (pJ/Spike/Synapse)	Synapse condition used	Synapse name/type
This work	1.05	250 kΩ–2.5 MΩ	Bi-Memristor
This work	2.4	15 kΩ–150 kΩ	Bi-Memristor
This work	8.1	2.5 kΩ–12 kΩ	Bi-Memristor
[6]	23.07	2 kΩ–10 kΩ with digital spike	Twin Memristor
[51]	9.3	1,000 synapses with 1 MΩ each	Resistive synapse
[20]	36.7	70 Ω–670 Ω	MD synapse
[5]	11–0.1	1 kΩ–1 MΩ	SDC memristor

than a corresponding integrate and fire neuron (employed in Reference [6]) implemented in the same technology [44] with a few picofarads of capacitance. With large crossbars and/or lower resistance synapses, the input current increases and therefore the capacitance must increase (to prevent disproportionate accumulation rate), further increasing the neuron's area. Also, as the neuron design changes with the device and system, the neuron's layout can change considerably, prompting the designer to alter the floorplan of the bigger system. Hence, the proposed design has the advantage of having a fixed and compact layout area (due to the minimum sized transistors used in the digital block).

In addition, the energy cost of the proposed design was evaluated in terms of the energy consumed per spike and is shown in Table 5 along with other values presented in literature. It is seen that the design in this work consumes less energy than most of those presented in the literature. In this table, it is noteworthy that the integrate and fire neuron of Reference [6], while using nearly the same synaptic resistance as the device Mem_1 here (2.5–12 kΩ) consumed more energy, demonstrating the energy efficiency of the proposed neuron.

8.3 Other Design Considerations

An advantage of using the constant (within a period) voltage-based spikes is that it is more robust to noise and is scaling friendly. As the neuromorphic crossbar size increases, the parasitic

resistance of the column increases. To compensate for this, width of metals in the routing can be increased. However, this increases the parasitic capacitance of the column (which is in addition to the capacitance contributed by the devices connected to the column). Hence, exponentially dampening or ramp-based analog spikes introduce the problem of charge-sharing between their spike generator circuit and the parasitic capacitances, which is eliminated by the constant voltage-based (coming from a supply) spike used here. Moreover, the WTA scheme used here introduces minimal overhead per neuron (Figure 14(b)), thus facilitating scaling without overhead concerns.

In addition to being amenable to scaling, the proposed design also helps eliminate sneak paths in a crossbar configuration by adopting a half-bias scheme. The proposed design here uses a VDD (positive)-VSS (negative) scheme, where GND is the mid-rail voltage. As explained in Section 4.2, the neuron's output (driving the rows) is at GND when it does not spike. Also, the columns are held at a virtual ground by the neurons' inputs. Hence, this puts the crossbar at a half-bias condition, reducing the sneak paths effect.

Therefore, the analysis presented in this work highlights the importance of considering the non-ideal behavior/properties of contemporary devices at the circuit design level for successful functioning of neuromorphic systems. The designs presented here and the analysis performed clearly demonstrate the immunity provided by them against device issues at the system level. Also, since the circuit parameters such as VDD, VSS, reference voltages, clock frequency, and duty cycle are tunable on-chip, they allow a systematic approach for the study of STDP. Based on any new experimental device behavior observed, the above circuits allow adaptation and hence a repetition of such experiments can be performed systematically without fundamentally altering our designs. Another important device-level issue is process variations that can impact the device properties. This issue must be addressed at the system's learning/operational algorithm level and hence falls out of scope of this work. In addition, CMOS circuits too suffer from process variations, which could impact the system. As a future work, the system presented here can be analysed for the impact of these variations. It may be noted that since the neuron presented here digitizes a range of input currents into a single value and performs a digital accumulation and comparison, it is expected to be robust under variations. As an extension of this work, designs presented here with their circuit level robustness can be used in conjunction with robust learning algorithms on deep neural networks to demonstrate more complex and robust machine learning tasks.

9 CONCLUSION

This work has demonstrated a device-aware circuit design methodology to mitigate the detrimental effects of device-level non-ideal behavior such as switching asymmetry, limited resolution, and endurance. Through efficient circuit design, careful control of STDP behavior has been demonstrated. Also, a generic neuron design has been proposed that can be used with a wide range of devices, hence precluding the need for re-design. The proposed designs have been employed together in a simple (single layer) pattern recognition system with STDP-based learning such that its performance is directly dependent on the learnt weights and hence on the efficiency of STDP. A scaling-friendly WTA scheme is also proposed for the system. It is shown that device-level issues can hamper the system's learning behavior and its performance. It is also demonstrated how the proposed designs can mitigate these adverse effects and make the system robust. Also, the area and energy costs of implementation of the proposed designs are discussed and it is shown that they are more efficient than those in the literature. These designs are also shown to be amenable to scaling, thereby indicating their suitability for use in bigger and more complex, yet robust, neuromorphic systems.

REFERENCES

[1] Shyam Prasad Adhikari, Changju Yang, Hyongsuk Kim, and Leon O. Chua. 2012. Memristor bridge synapse-based neural network and its learning. *IEEE Trans. Neural Netw. Learn. Syst.* 23, 9 (2012), 1426–1435.

[2] Fabien Alibart, Elham Zamanidoost, and Dmitri B. Strukov. 2013. Pattern classification by memristive crossbar circuits using ex situ and in situ training. *Nature Commun.* 4 (2013), 2072.

[3] Sherif Amer, Sagarvarma Sayyaparaju, Garrett S. Rose, Karsten Beckmann, and Nathaniel C. Cady. 2017. A practical hafnium-oxide memristor model suitable for circuit design and simulation. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'17)*. IEEE, 1–4.

[4] Karsten Beckmann, Josh Holt, Harika Manem, Joseph Van Nostrand, and Nathaniel C. Cady. 2016. Nanoscale hafnium oxide rram devices exhibit pulse dependent behavior and multi-level resistance capability. *Mrs Adv.* 1, 49 (2016), 3355–3360.

[5] Kristy A. Campbell, Kolton T. Drake, and Elisa H. Barney Smith. 2016. Pulse shape and timing dependence on the spike-timing dependent plasticity response of ion-conducting memristors as synapses. *Front. Bioengineer. Biotechnol.* 4 (2016), 97.

[6] Gangotree Chakma, Md Musabbir Adnan, Austin R. Wyer, Ryan Weiss, Catherine D. Schuman, and Garrett S. Rose. 2018. Memristive mixed-signal neuromorphic systems: Energy-efficient learning at the circuit-level. *IEEE J. Emerg. Select. Top. Circ. Syst.* 8, 1 (2018), 125–136.

[7] B. Chen, Y. Lu, B. Gao, Y. H. Fu, F. F. Zhang, P. Huang, Y. S. Chen, L. F. Liu, X. Y. Liu, J. F. Kang, et al. 2011. Physical mechanisms of endurance degradation in TMO-RRAM. In *Proceedings of the International Electron Devices Meeting*. IEEE, 12–3.

[8] Ling Chen, Chuandong Li, Tingwen Huang, Yiran Chen, and Xin Wang. 2014. Memristor crossbar-based unsupervised image learning. *Neural Comput. Appl.* 25, 2 (2014), 393–400.

[9] Tanguy Chouard and Liesbeth Venema. 2015. Machine intelligence. https://www.nature.com/articles/521435a.

[10] Myonglae Chu, Byoungho Kim, Sangsu Park, Hyunsang Hwang, Moongu Jeon, Byoung Hun Lee, and Byung-Geun Lee. 2015. Neuromorphic hardware system for visual pattern recognition with memristor array and CMOS neuron. *IEEE Trans. Industr. Electron.* 62, 4 (2015), 2410–2419.

[11] Leon Chua. 1971. Memristor-the missing circuit element. *IEEE Trans. Circ. Theory* 18, 5 (1971), 507–519.

[12] Erika Covi, Stefano Brivio, Alexander Serb, Themis Prodromakis, Marco Fanciulli, and Sabina Spiga. 2016. Analog memristive synapse in spiking networks implementing unsupervised learning. *Front. Neurosci.* 10 (2016), 482.

[13] Erika Covi, Stefano Brivio, Alexantrou Serb, Themistoklis Prodromakis, M. Fanciulli, and S. Spiga. 2016. HfO2-based memristors for neuromorphic applications. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'16)*. IEEE, 393–396.

[14] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. Retrieved from http://archive.ics.uci.edu/ml.

[15] H. P. Graf, L. D. Jackel, R. E. Howard, B. Straughn, J. S. Denker, W. Hubbard, D. M. Tennant, and D. Schwartz. 1986. VLSI implementation of a neural network memory with several hundreds of neurons. In *Proceedings of the American Institute of Physics Conference*, Vol. 151. AIP, 182–187.

[16] Mirko Hansen, Finn Zahari, Martin Ziegler, and Hermann Kohlstedt. 2017. Double-barrier memristive devices for unsupervised learning and pattern recognition. *Front. Neurosci.* 11 (2017), 91.

[17] Raqibul Hasan, Tarek M. Taha, and Chris Yakopcic. 2017. On-chip training of memristor crossbar based multi-layer neural networks. *Microelectron. J.* 66 (2017), 31–40.

[18] Nabeem Hashem and Shamik Das. 2012. Switching-time analysis of binary-oxide memristors via a nonlinear model. *Appl. Phys. Lett.* 100, 26 (2012), 262106.

[19] Wei He, Kejie Huang, Ning Ning, Kiruthika Ramanathan, Guoqi Li, Yu Jiang, JiaYin Sze, Luping Shi, Rong Zhao, and Jing Pei. 2014. Enabling an integrated rate-temporal learning scheme on memristor. *Sci. Rep.* 4 (2014), 4755.

[20] Miao Hu, Yiran Chen, J. Joshua Yang, Yu Wang, and Hai Helen Li. 2017. A compact memristor-based dynamic synapse for spiking neural networks. *IEEE Trans. Comput.-Aid. Design Integr. Circ. Syst.* 36, 8 (2017), 1353–1366.

[21] Giacomo Indiveri, Bernabé Linares-Barranco, Tara Julia Hamilton, André Van Schaik, Ralph Etienne-Cummings, Tobi Delbruck, Shih-Chii Liu, Piotr Dudek, Philipp Häfliger, Sylvie Renaud, et al. 2011. Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5 (2011), 73.

[22] Doo Seok Jeong, Kyung Min Kim, Sungho Kim, Byung Joon Choi, and Cheol Seong Hwang. 2016. Memristors for energy-efficient new computing paradigms. *Adv. Electron. Mater.* 2, 9 (2016), 1600090.

[23] Li Jiang, Fu-Cheng Lv, Rui Yang, Dan-Chun Hu, and Xin Guo. 2018. Forming-free artificial synapses with Ag point contacts at interface. *J. Materiom.* (2018).

[24] Sung Hyun Jo, Ting Chang, Idongesit Ebong, Bhavitavya B. Bhadviya, Pinaki Mazumder, and Wei Lu. 2010. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 4 (2010), 1297–1301.

[25] Hyongsuk Kim, Maheshwar Pd Sah, Changju Yang, Tamás Roska, and Leon O. Chua. 2012. Neural synaptic weighting with a pulse-based memristor circuit. *IEEE Trans. Circ. Syst. I: Reg. Papers* 59, 1 (2012), 148–158.

[26] Sungho Kim, Meehyun Lim, Yeamin Kim, Hee-Dong Kim, and Sung-Jin Choi. 2018. Impact of synaptic device variations on pattern recognition accuracy in a hardware neural network. *Sci. Rep.* 8, 1 (2018), 2638.

[27] Gwendal Lecerf, Jean Tomas, and Sylvain Saïghi. 2013. Excitatory and inhibitory memristive synapses for spiking neural networks. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'13)*. IEEE, 1616–1619.

[28] Can Li, Daniel Belkin, Yunning Li, Peng Yan, Miao Hu, Ning Ge, Hao Jiang, Eric Montgomery, Peng Lin, Zhongrui Wang, et al. 2018. Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. *Nature Commun.* 9, 1 (2018), 2385.

[29] Bernabe Linares-Barranco, Teresa Serrano-Gotarredona, Luis A Camuñas-Mesa, Jose A. Perez-Carrasco, Carlos Zamarreño-Ramos, and Timothee Masquelier. 2011. On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. *Front. Neurosci.* 5 (2011), 26.

[30] Ke Lu, Yi Li, Wei-Fan He, Jia Chen, Ya-Xiong Zhou, Nian Duan, Miao-Miao Jin, Wei Gu, Kan-Hao Xue, Hua-Jun Sun, et al. 2018. Diverse spike-timing-dependent plasticity based on multilevel HfO x memristor for neuromorphic computing. *Appl. Phys. A* 124, 6 (2018), 438.

[31] Mohammad Maymandi-Nejad and Manoj Sachdev. 2003. A digitally programmable delay element: Design and analysis. *IEEE Trans. Very Large Scale Integr. Syst.* 11, 5 (2003), 871–878.

[32] Saber Moradi and Giacomo Indiveri. 2014. An event-based neural network architecture with an asynchronous programmable synaptic memory. *IEEE Trans. Biomed. Circ. Syst.* 8, 1 (2014), 98–107.

[33] Takayuki Morishita, Youichi Tamura, Tatsuo Otsuki, and Gota Kano. 1992. A BiCMOS analog neural network with dynamically updated weights. *IEICE Trans. Electron.* 75, 3 (1992), 297–302.

[34] Yu Nishitani, Yukihiro Kaneko, and Michihito Ueda. 2015. Supervised learning using spike-timing-dependent plasticity of memristive synapses. *IEEE Trans. Neural Netw. Learn. Syst.* 26, 12 (2015), 2999–3008.

[35] Sangsu Park, Myonglae Chu, Jongin Kim, Jinwoo Noh, Moongu Jeon, Byoung Hun Lee, Hyunsang Hwang, Boreom Lee, and Byung-geun Lee. 2015. Electronic system with memristive synapses for pattern recognition. *Sci. Rep.* 5 (2015), 10123.

[36] Melika Payvand, Justin Rofeh, Avantika Sodhi, and Luke Theogarajan. 2014. A CMOS-memristive self-learning neural network for pattern classification applications. In *Proceedings of the IEEE/ACM International Symposium on Nanoscale Architectures*. ACM, 92–97.

[37] G. Pedretti, V. Milo, S. Ambrogio, R. Carboni, S. Bianchi, A. Calderoni, N. Ramaswamy, A. S. Spinelli, and D. Ielmini. 2017. Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity. *Sci. Rep.* 7, 1 (2017), 5288.

[38] Peyman Pouyan, Esteve Amat, and Antonio Rubio. 2015. Statistical lifetime analysis of memristive crossbar matrix. In *Proceedings of the 10th International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS'15)*. IEEE, 1–6.

[39] Mirko Prezioso, Farnood Merrikh-Bayat, B. D. Hoskins, Gina C. Adam, Konstantin K. Likharev, and Dmitri B. Strukov. 2015. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521, 7550 (2015), 61.

[40] Damien Querlioz, W. S. Zhao, Philippe Dollfus, J.-O. Klein, Olivier Bichler, and Christian Gamrat. 2012. Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches. In *Proceedings of the IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH'12)*. IEEE, 203–210.

[41] Shubha Ramakrishnan, Paul E. Hasler, and Christal Gordon. 2011. Floating gate synapses with spike-time-dependent plasticity. *IEEE Trans. Biomed. Circ. Syst.* 5, 3 (2011), 244–252.

[42] Vishal Saxena, Xinyu Wu, Ira Srivastava, and Kehan Zhu. 2018. Towards neuromorphic learning machines using emerging memory devices with brain-like energy efficiency. *J. Low Power Electron. Appl.* 8, 4 (2018), 34.

[43] Sagarvarma Sayyaparaju, Sherif Amer, and Garrett S. Rose. 2018. A bi-memristor synapse with spike-timing-dependent plasticity for on-chip learning in memristive neuromorphic systems. In *Proceedings of the 19th International Symposium on Quality Electronic Design (ISQED'18)*. IEEE, 69–74.

[44] Sagarvarma Sayyaparaju, Ryan Weiss, and Garrett S. Rose. 2018. A mixed-mode neuron with on-chip tunability for generic use in memristive neuromorphic systems. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'18)*. IEEE, 441–446.

[45] Teresa Serrano-Gotarredona, Timothée Masquelier, Themistoklis Prodromakis, Giacomo Indiveri, and Bernabe Linares-Barranco. 2013. STDP and STDP variations with memristors for spiking neuromorphic learning systems. *Front. Neurosci.* 7 (2013), 2.

[46] Ahmad Muqeem Sheri, Hyunsang Hwang, Moongu Jeon, and Byung-geun Lee. 2014. Neuromorphic character recognition system with two PCMO memristors as a synapse. *IEEE Trans. Industr. Electron.* 61, 6 (2014), 2933–2941.

[47] Patrick Sheridan, Wen Ma, and Wei Lu. 2014. Pattern recognition with memristor networks. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'14)*. IEEE, 1078–1081.

[48] Greg S. Snider. 2008. Spike-timing-dependent learning in memristive nanodevices. In *Proceedings of the IEEE International Symposium on Nanoscale Architectures*. IEEE Computer Society, 85–92.
[49] Dmitri B. Strukov, Gregory S. Snider, Duncan R. Stewart, and R. Stanley Williams. 2008. The missing memristor found. *Nature* 453, 7191 (2008), 80.
[50] Yi Sun, Hui Xu, Chao Wang, Bing Song, Haijun Liu, Qi Liu, Sen Liu, and Qingjiang Li. 2018. A Ti/AlO x/TaO x/Pt analog synapse for memristive neural network. *IEEE Electron. Device Lett.* 39, 9 (2018), 1298–1301.
[51] Xinyu Wu, Vishal Saxena, Kehan Zhu, and Sakkarapani Balagopal. 2015. A CMOS spiking neuron for brain-inspired neural networks with resistive synapses andIn SituLearning. *IEEE Trans. Circ. Syst. II: Express Briefs* 62, 11 (2015), 1088–1092.
[52] J. Joshua Yang, M.-X. Zhang, John Paul Strachan, Feng Miao, Matthew D. Pickett, Ronald D. Kelley, G. Medeiros-Ribeiro, and R. Stanley Williams. 2010. High switching endurance in TaO x memristive devices. *Appl. Phys. Lett.* 97, 23 (2010), 232102.
[53] Peng Yao, Huaqiang Wu, Bin Gao, Sukru Burc Eryilmaz, Xueyao Huang, Wenqiang Zhang, Qingtian Zhang, Ning Deng, Luping Shi, H.-S. Philip Wong, et al. 2017. Face classification using electronic synapses. *Nature Commun.* 8 (2017), 15199.
[54] Finn Zahari, Mirko Hansen, Thomas Mussenbrock, Martin Ziegler, and Hermann Kohlstedt. 2015. Pattern recognition with TiOx-based memristive devices. *AIMS Mater. Sci.* 2 (2015), 203–216.
[55] Errui Zhou, Liang Fang, and Binbin Yang. 2018. Memristive spiking neural networks trained with unsupervised STDP. *Electronics* 7, 12 (2018), 396.

Received May 2019; revised November 2019; accepted January 2020