A Real-time Trainable and Clock-less Spiking Neural Network with 1R Memristive Synapses

Aditya Shukla and Udayan Ganguly, Member, IEEE

Abstract—Owing to their unexplored but potentially superior computational capability and remarkably low power consumption for executing brain-like tasks, spiking neural networks (SNNs) have come under the scope of several research groups. For modeling a network of spiking neurons and synapses highly parallelized hardware with a distributed and localized processor and memory is essential. Highly integrable resistive RAM or ReRAM, with voltage-pulse moldable analog conductivity, has a potential to play a key role in such a hardware model, as it can and has been shown to mimic a synapse and its time dependent plasticity. Being a two terminal device, however, it’s not directly suitable for naturally/biologically asynchronous SNNs because it requires ability to do reading and writing concurrently. In this work, we overcome this challenge and present a clock-less approach to implement both learning and recognition on a spiking neural network with memristive synaptic array wherein, reading and writing are performed in different frequency domains. Further, we validate our scheme in SPICE by translating a mathematical mono-layered feed-forward Iris classifying SNN. A hardware implementation of the scheme will eliminate the need for clocks and switches to drive learning and has the potential to be used in the development of real-time trainable hardware recurrent networks.

Index Terms—Artificial brain, clock-less neural networks, event-based neuromorphic system, Fisher-Iris classifier, memristors, neuromorphic engineering, resistive RAM, spiking neural networks, synaptic time-dependent plasticity.

I. INTRODUCTION

Since 1950s, large effort has been put into building automated electronic systems that are faster and more capable than humans, motivated by both curiosity and commercial interests. Most obvious such systems are von Neumann-architecture based computers, in which complicated tasks are broken down into simple instructions, executed sequentially. However, for intelligent tasks in which our brain excels like perception, decision making, learning in-situ and recognition, such systems are fundamentally lacking.

Inspired from brain, a fairly recent approach is to compute using spatio-temporal voltage spiking patterns [1][2][3]. Since all the information in voltage-spike-timings, a direct implementation of such a system should consume very little power. Powerful Neumann computers have been used to perform these tasks and such attempts have been successful to some extent, but are not as power or area efficient as our brain [4][5]. Since last decade, large-scale non-von Neumann electronic models of brain have been implemented with silicon neurons, embedded pre-programmed memory acting as synapse and additional complex communication circuitry [6][7][8]. A few VLSI implementations that can also be trained in-situ make use of either multi-bit SRAMs [9] or custom circuit-based analog memories [10], which may not be efficient use of area. Secondly, these implementations too, make use of complex communication schemes for inter-neuron communication and training that require additional power and area. One of the major roadblocks for full-scale implementation and utilization of a network of spiking neurons is the lack of a general training method. In brain, synaptic time dependent plasticity (STDP) was shown as an unsupervised training mechanism in which the change in weight depends on the difference of spiking-times of a pre and post-synaptic neuron [11]. Other methods of training [12]–[14] are either directly derived from the domain of artificial neural networks – thus out of our scope – or, are not easy to implement on hardware on a large-scale.

Resistive memories (a.k.a. memristors or ReRAMS) are extremely compact, non-volatile passive memory devices composed of layers of transition-metal oxides sandwiched between two metallic cross-bars that serve as its two terminals. These not only have great potential in memory applications but also in non-conventional computing techniques like neural networks [15]–[17] and stochastic computing [18]. Dense array of these elements can serve as an either embedded or chip-based synaptic-array, with a neuron’s axon connected to one terminal of a device and dendrite of the other neuron to the second [17].

In the simplest of all ReRAM based neural networks, each element of the memory directly acts as weight. A common way to modify these weights, according to the STDP rule, is to use special voltage waveforms derived directly from it [19] (See Figure 1). These waveforms are generated by a neuron every time it spikes. Difference between timings of two spikes is translated into a voltage pulse, which, if above the threshold of the ReRAM, changes its conductance. This way STDP has been demonstrated in various ReRAM devices. System level demonstration spiking neural networks of such a technique,
however, are very few primarily because ReRMs are very not reliable and deterministic.

A source of dilemma for using bi-terminal ReRMs for on-field learning applications in SNNs is that reading and writing has to be done simultaneously and arbitrarily in time. Reading is generally done with long, time-varying pulses and writing is done with equally long pulses with waveform shown in Figure 1. Read and write process will disturb each other, if both read and write voltages are simultaneously applied simultaneously on top of each other. In [9], [10] read and write were time-multiplexed, [20]–[22] suggested switching off the reading process when writing occurs, which may not be general enough. In [23] use of two arrays, one for reading and one for writing was suggested. Write array’s content was periodically copied into the read array.

In this paper, we overcome this issue by using a scheme inspired from frequency division multiplexing (FDM). FDM is commonly used for parallel communication of information from multiple independent sources, through multiple non-overlapping frequency bands. We first set up a general software framework that suffices for simple SNN hardware implementations. Next we elucidate the dilemma that we previously introduced, followed by our proposal to resolve it. Then we report our demonstration work involving SPICE descriptions of SNN components and an Iris classifier network, in details. Lastly, we explore the effects of a non-linear ReRAM on out networks performance.

II. SPIKING NEURAL NETWORK – SOFTWARE FRAMEWORK

A software model SNN consists of two essential elements: (1) neuron model and (2) interconnecting synapse. We now briefly describe all these components, taking a mono-layered feed-forward network with m input neurons connected to n output neurons through a network on m × n synaptic array as a reference.

1) Neuron

Several models of neurons exist, simplest being the leaky-integrate and fire (LIF) [3]. The LIF neuron state variable varies in time as Eq. 1, where, V(t) is the hidden state variable, s(t) is a V(t)-dependent visible binary variable, R and C are neuron-specific parameters and I(t) is the excitation sources. Whenever the state variable V(t) reaches a constant threshold (Vt), LIF is defined to be in a spiking state, s(t) is momentarily pulsed and V(t) is reset to its initial value. After spiking, LIF goes to a ‘sleeping’ state for a fixed refractory period and the input excitation signal is unable to cause any change in the neuron’s state.

\[ C \frac{dV}{dt} = \frac{V(t)}{R} + \sum I_{a,j} \star f(t) \]  

(1)

2) Synapse and synaptic time-dependent plasticity

Synapse generates a time varying current in response to its source (pre-)- neuron’s spike, which may encourage/inhibit spiking of the sink (post-) synaptic neuron (See Eq. 2). This current is proportional to the strength/weight (wij) of a synapse between i\textsuperscript{th} pre-neuron and j\textsuperscript{th} post neuron.

\[ I_{a,j} = w_{ij} \alpha_{t_1,t_2}(t) \]  

(2)

Where,

\[ \alpha_{t_1,t_2}(t) = V_0 \left( \exp \left( \frac{-t}{t_1} \right) - \exp \left( \frac{-t}{t_2} \right) \right) \]  

(3)

Here, V0, t1 and t2 are network design parameters. Depending on the timings of the pre- and post-synaptic neurons, the synaptic weight changes according to a network specific rule. For all purposes, STDP rule given in Eq. 3 has been used.
\[ \Delta w = \begin{cases} |A_+| \exp \left( -\frac{\Delta t}{\tau_+} \right), & \Delta t \geq 0 \\ -|A_-| \exp \left( -\frac{\Delta t}{\tau_-} \right), & \Delta t < 0 \end{cases} \]  

Here, \( A_+ \), \( A_- \), \( \tau_+ \) and \( \tau_- \) are network design parameters.

III. SIMULTANEOUS READ-WRITE DILEMMA

The equivalent hardware of the SNN discussed above needs to (1) read the synaptic-memory whenever neurons – belonging to layers other than the input – have to be driven and (2) change in its synaptic-memory each time spiking occurs. The read process involves application of several synaptic spike response voltage-pulse (as per equation X; called read-pulse henceforth) at the axon-terminal of the ReRAM mimicking a synapse \( w_{ij} \) by neuron \( N_i \). This leads to a current into neuron \( N_j \) at its virtually grounded dendron-terminal, proportional to the conductance of the ReRAM. Synaptic weights are changed by long write pulses that begin with a brief writing-spike and immediately followed by a slower time-keeping pulse. This sharp or brief pulse applied by the neuron at one end of the neuron changes the conductance of ReRAM, whenever applied along with a timer pulse of another neuron on the opposite end of the ReRAM.

Thus both processes occurring in parallel involve application of voltage pulses. If one were to simplistically apply a superposition of these pulses to carry out the processes in parallel, either processes will get drastically disturbed. To mitigate this problem, either the synaptic structure/device or nature of pulses being used to do these processes may be changed. In the subsequent sections, we suggest and demonstrate a technique to mitigate this issue. We change the nature of the pulses as we perform the reading process in a higher frequency domain.

IV. PROPOSAL FOR SOLVING SIMULTANEOUS READ-WRITE DILEMMA

To perform read and write concurrently on a ReRAM array, we propose a technique inspired from frequency division multiplexing (FDM) used in several communication systems. To elucidate the proposal, we split it into three steps: (A) reduction of read-pulse width, (B) Use of high frequency sinusoidal spikes for reading and (C) Read-current extraction.

A. Reduction of read-pulse’s width

Assuming that the weights evolve slowly (e.g. less than 1% change within a read-pulse [24]), we may treat the weighing process to be linear and time-invariant (more accurately, quasi-LTI). Note that the generation of the read pulses from the brief spikes of the neurons (See Eq. 2) is an LTI process as well. Given two or more LTI systems connected in series, inter-changing their order does not affect the input-output behavior of the containing system. Thus, we can inter-change the read-pulse generator with Re-RAM weighing array. This directly connects spiking/pulsing neurons to the array. This it is now possible to use neuron’s brief spikes for the reading process and alleviate the disturbance caused to the writing process. However extraction of branch current due to the read pulses is still impossible, thus necessitating the next step.

B. Use of high frequency sinusoidal spikes for reading

Instead of using conventional rectangular voltage pulses to do the reading, we propose the use of sinusoids, which means executing the reading process in a higher frequency domain. Each time a neuron spikes, a sinusoidal waveform to axonal-end of the ReRAM is applied. This waveform should be long and strong enough so that the current though the ReRAMs can be sensed at the array output, but too short and weak to cause any significant change in the conductance.

C. Read-current extraction

After applying the modifications suggested in steps A-B, synaptic reading process can be approximately regarded as an instantaneous process wherein each pre-neural spike is weighed at the spiking instant and is fed-forward to drive subsequent neurons. In our implementation, for each spike that is a fixed amplitude, high frequency, voltage-sinusoid, weighing is manifested by ohm’s law based currents, thus giving a sinusoidal current proportional to the dynamic conductance (equal to the absolute conductance for linear ReRAMs). Since the reading is occurring in a different frequency domain, a filter has to be used to extract the sinusoidal component and diminish that of the write-pulse. The output of the filter will oscillate around zero and may be rectified, thus extracting the amplitude. This rectified signal drives the (LTI) synaptic spike response generator, whose output then drives the LIF neurons’ parallel-RC core.
For the scheme to work out as expected, the ReRAM being employed as a synapse needs to have a linear current-voltage relationship. However most ReRAMs developed so far are either inherently or by virtue of engineering, non-linear. For a linear ReRAM, the dynamic conductance is a constant with bias and is same as the absolute conductance of the device. For a quadratic ReRAM, the dynamic conductance varies: first nearly constant then linearly with the bias voltage (See Figure 12). Thus, even if the states of the synaptic ReRAM for two space- and/or time-separated spikes are different, the current fed-forward may be same. Simply put, the effective conductance depends on bias. We explore the effects of such a device on an SNN in a subsequent section.

V. SCHEME’S DEMONSTRATION AND RESULTS

We first briefly describe, demonstrate the working of and validate the (1) individual components, then (2) a 1×1 network and finally (3) a 16×3 Iris flower classifying SNN.

A. Components of the network

The SPICE network consisted of models of (1) LIF neuron (2) ReRAM as a synapse (3) write-pulse generator (4) read-pulse generator and (4) high-pass filter. Now we describe the individual components:

1) LIF Neurons

An LIF neuron model has been described using a resistor, a capacitor and a capacitor-reset switch. A refractory unit has also been added that blocks the current input for some controllable time after each spiking event.

To validate the neurons functioning, we compared the SPICE neuron’s spiking times with that of the software neuron with same current input. We used several rectangular current pulses with a time-length of 100ms and uniformly distributed in the range 0 to 20nA, for driving the test-neuron having a refractory time of 5ms. After completion of each of the 100 current-pulses, the neuron was reset, regardless of its internal state. Note that this driving input is similar to the input being provided to the neuron in the network’s software implementation.

It was observed that SPICE neurons performed almost same as the software neuron. This can be seen from the Figure 6 which shows the distribution of the spike-jitter. Most of the jitter is close to the ideal value of 0ms. The maximum jitter value of 2ms is much less than minimum inter-spike interval (17ms). Since the final performance is unaffected, we assume the jitter to be negligibly small.

![Figure 5](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

![Figure 6](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

![Figure 7](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

![Figure 8](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

![Figure 9](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

V. SCHEME’S DEMONSTRATION AND RESULTS

We first briefly describe, demonstrate the working of and validate the (1) individual components, then (2) a 1×1 network and finally (3) a 16×3 Iris flower classifying SNN.

A. Components of the network

The SPICE network consisted of models of (1) LIF neuron (2) ReRAM as a synapse (3) write-pulse generator (4) read-pulse generator and (4) high-pass filter. Now we describe the individual components:

1) LIF Neurons

An LIF neuron model has been described using a resistor, a capacitor and a capacitor-reset switch. A refractory unit has also been added that blocks the current input for some controllable time after each spiking event.

To validate the neurons functioning, we compared the SPICE neuron’s spiking times with that of the software neuron with same current input. We used several rectangular current pulses with a time-length of 100ms and uniformly distributed in the range 0 to 20nA, for driving the test-neuron having a refractory time of 5ms. After completion of each of the 100 current-pulses, the neuron was reset, regardless of its internal state. Note that this driving input is similar to the input being provided to the neuron in the network’s software implementation.

It was observed that SPICE neurons performed almost same as the software neuron. This can be seen from the Figure 6 which shows the distribution of the spike-jitter. Most of the jitter is close to the ideal value of 0ms. The maximum jitter value of 2ms is much less than minimum inter-spike interval (17ms). Since the final performance is unaffected, we assume the jitter to be negligibly small.

![Figure 5](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

![Figure 6](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

![Figure 7](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

![Figure 8](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

![Figure 9](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAIoAAADcCAIAAADxgt35AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyXRFWHRMCQ4APJ refere...)

V. SCHEME’S DEMONSTRATION AND RESULTS

We first briefly describe, demonstrate the working of and validate the (1) individual components, then (2) a 1×1 network and finally (3) a 16×3 Iris flower classifying SNN.

A. Components of the network

The SPICE network consisted of models of (1) LIF neuron (2) ReRAM as a synapse (3) write-pulse generator (4) read-pulse generator and (4) high-pass filter. Now we describe the individual components:

1) LIF Neurons

An LIF neuron model has been described using a resistor, a capacitor and a capacitor-reset switch. A refractory unit has also been added that blocks the current input for some controllable time after each spiking event.

To validate the neurons functioning, we compared the SPICE neuron’s spiking times with that of the software neuron with same current input. We used several rectangular current pulses with a time-length of 100ms and uniformly distributed in the range 0 to 20nA, for driving the test-neuron having a refractory time of 5ms. After completion of each of the 100 current-pulses, the neuron was reset, regardless of its internal state. Note that this driving input is similar to the input being provided to the neuron in the network’s software implementation.

It was observed that SPICE neurons performed almost same as the software neuron. This can be seen from the Figure 6 which shows the distribution of the spike-jitter. Most of the jitter is close to the ideal value of 0ms. The maximum jitter value of 2ms is much less than minimum inter-spike interval (17ms). Since the final performance is unaffected, we assume the jitter to be negligibly small.
2) **ReRAM as a synapse**

This model acts as a resistive memory via an ideal capacitor, whose charge determines its two independent state variables – conductivity and write-threshold. This charge can be modified via application of voltage pulses that are higher than the write-threshold. The software-equivalent ideal ReRAM was used for software-equivalence validation of the network that had a state-independent writing threshold. Then, a realistic ReRAM model (of the device in [25]) was used to evaluate effect of realistic ReRAM features (e.g. asymmetry in potentiation vs. depression) on the learning performance. Lastly, the effect of non-linear ReRAMs on the network’s general performance was studied by replacing the linear dependent-conductance with non-linear one.

3) **Write-pulse generator**

This unit is a two-pole linear time-invariant (LTI) system with a resetting switch. The output potential of this unit is forced to a constant positive voltage with a switch whenever a spike (which is a rectangular voltage pulse) is applied at the input. As soon as the input pulse is removed, the output potential drops to an opposite polarity and subsequently decays (Figure 1f). A new spike would immediately reset and trigger a new waveform.

4) **Synaptic spike response (α-function) generator**

This unit is another LTI system with impulse response given by the α-function in Eq. 2-3 (The time constants involved in this equation are much longer than driving spike/pulse width, thus the spikes/pulses can be approximated as an random impulse train.) This unit gives sum of time-shifted alpha function in response to incoming spikes (or, approximately a train of delta functions).

5) **High-pass filter**

To extract the high-frequency read-pulse current from the sensed branch current, we described and used a behavioral model of a high-pass filter with cut-off frequency $\omega_C$ (See Figure 3).

### B. Neuron-synapse-neuron demonstration

Being the simplest possible network that uses all the components described designed/described in SPICE, we first validate a 1 by 1 array consisting of an input neuron driving another neuron connected to it via a ReRAM based synapse with STDP implemented using write-pulse generator driven using filtered spikes. This was done by comparing reading and writing processes separately.

We evaluate the reading process by comparing the filter output that is ideally proportional to the sampled conductance at the spiking instant and the actual conductance signal. Histogram in Figure 5 shows the distribution of percentage error. The maximum error is around 2.0% which corroborates our SPICE description.

To validate the write-process, we randomly excite a presynaptic neuron that drive another neuron via ReRAM synapse and noted the change in conductance for every pair of pre and post spikes. It is seen that intended change in the conductance, as a function of pre-post spike-time-difference, is achieved via the STDP-waveform-pair.

### C. Iris classifier network with an ideal synaptic model

For validating the proposed scheme, we implemented the feed-forward SNN Iris classifier network reported in [24] using NG-SPICE (version-25). A small part of Fisher’s Iris data is first applied several times for training by the STDP rule and then for testing the classification. An all-sample accuracy

---

**Figure 10:**

(a) Reading process involves signal flow from first neural layer to the next via weighing synaptic array (b) Writing process (concurrent in time with reading) involves application of writing pulses from both sides (c) Flowchart of the signals along with the processing units for a simple implementation using suggested one-array scheme; (1) Input being provided from the sensor which drives the LIF unit; (2) Voltage is spiked each time membrane potential crosses the threshold; (3) a “time since last spike” dependent waveform is generated each time spiking occurs. To this waveform, brief fast-varying sinusoid voltage is added; (4) Current through the current sense resistance; (5) High pass filter followed by an envelope detector transform sinusoids into rectangular voltage pulses; (6) These rectangular pulses drive the α-function generator, which then drives the second layer LIF unit; (7) The second layer neuron undergoes similar process as the first layer neurons, generating write-pulses shown in (8); (9) ReRAM’s weight changes whenever the voltage across it crosses its threshold.
of 97.33\% is achieved by applying four linear transformations to the raw data, giving a network having a size of 16×3.

We first directly translated the mathematical synapse into SPICE, as a ReRAM (henceforth called an ideal ReRAM). Figure 7 compares the essential transients associated with the first synapse of the network for a random sample, with an ideal synaptic ReRAM. A lower weight change is observed due to the smoothing of the waveforms but that, however, only lowers the learning rate but not the learning performance. Network’s learning performance for an ideal ReRAM is given in Figure 8. After training for twenty epochs for various initial weight configurations, we are able to achieve classification accuracy greater than 90%.

D. Iris classifier network with HfO\textsubscript{2} memristive synaptic model

A somewhat realistic model of HfO\textsubscript{2} ReRAM was developed by the matching DC I-Vs and an inferior performance is observed (See Figure 11). In fact, the weights and the classification accuracy did not stabilize. The maximum and average classification accuracies for both ReRAM models are compared in Table 1.

VI. EFFECT ON PERFORMANCE DUE TO I-V NON-LINEARITY

For non-linear devices, the filtered branch-current depends on the state of the devices and the bias at which the sinusoid was applied (Bias being the net write-pulses’ voltage that appeared across the devices at the spiking instant.) More compactly, the current and hence the driving signal depends on the dynamic conductance.

To explore the effect of bias dependence of the dynamic conductance, we replace the following read equation in our algorithm:

\[ I(t) = w_{ij} \alpha_{v_1,v_2}(t) \]

with:

\[ I(t) = w_{ij} G_{read}(v_{bias}, A) \alpha_{v_1,v_2}(t) \]

where, \( G_{read}(v_{bias}, A) \) is the bias and amplitude dependent normalized dynamic conductance of an ideal ReRAM and is plotted in Figure 12. Figure 13 plots the classification accuracy of the network for various amplitudes as learning progresses. Though learning improves as amplitudes are increased, the writing process may get affected. Figure 14 plots the initial and final states for an ideal linear and quadratic ReRAM (The state for a latter is defined uniquely by reading the current at a fixed voltage). We see that final weight configuration is similar for all cases suggesting that the weight evolution is on the right track.

VII. BENCHMARKING

Several authors have reported their electronic implementation of biologically inspired neural network using both S- and Re-RAM arrays. Most these works are discrete time systems and spiking is synchronized with a common clock. Although some groups also focus on arbitrarily timed spiking or asynchronous networks, none of their work talks about concurrent read-write dilemma in particular. In [26],

| Table I: Performance Summary |
|-----------------------------|
| RRAM I-V type | Amplitude (mA) | Classification accuracy |
| Linear         | 0.01           | 97.3\%                |
|                | 0.1            |                       |
| Quadratic      | 0.01           | 91.3\%                |
|                | 0.1            | 93.0\%                |
|                | 0.3            | 96.0\%                |

Figure 11: (left) Measured DC-IV of HfO\textsubscript{2} ReRAM device and simulated DV-IV of device’s behavioural model; (right) Percentage accuracy for Iris-classifying SNN using the model, as training progresses.

Figure 12: (left) Normalized dynamic conductance for quadratic ReRAM model brought to a fixed iso-voltage DC (static) conductance. The y-intercept as read-amplitude is increased; (right) Distribution of bias voltage for the first epoch. Similar distribution peaking at 0V is observed (not shown) for subsequent epochs.

Figure 13: Percentage recognition increases as the number of iterations increases.

Figure 14: Initial and final (45 epochs) iso-voltage conductance maps for linear and quadratic ReRAM models. It is seen that if sinusoid’s amplitude is big enough, final weights approach similar values regardless of the ReRAM’s I-V type.
authors mitigate this problem by switching-off the current integration in post-synaptic neurons each time a pre-synaptic neuron spikes. However, (1) this requires additional communication channel between layers of neurons and (2) if the number of pre-neurons is large, integration at the post-layer may get affected. In [23], we split reading and writing processes physically by using two ReRAM arrays – one array is used for reading and the other for writing. As learning progresses, write array is written onto and periodically, write array’s content is transferred onto the read array. This requires an additional circuitry to transfer weight and thus is not really asynchronous.

Our scheme, if implemented with due device and network considerations, will yield a clock-less, switch-less and arbitrarily timed spiking neural network with potential to be recurrent (feed-back) with any number of layers.

VIII. CONCLUSION

For hardware SNNs employing cross-bar ReRAM arrays as synaptic-array, we propose performing read and writing processes in different frequency domains to be able to do them concurrently. To demonstrate our idea, we translate a mathematical SNN based Iris-classifier into SPICE and successfully achieve mathematical training performance (i.e., 97.3% accuracy). The demo is split in three independent parts. First a mathematical synapse’s is directly translated to SPICE and used as an ideal ReRAM. To explore/test the effect of using real ReRAMs, we simulated our SPICE network using partly realistic model of HfO2 ReRAM by matching I-Vs. After training, the learning performance was found to degrade to a maximum of 86% accuracy. Finally, to explore the effect of more common non-linear ReRAMs, we model these as symmetric devices with a state-independent threshold. It is observed that it performs similarly as a linear and constant threshold ReRAM, and gives better results as sinusoid amplitudes are increased.

REFERENCES

[1] J. Von Neumann, The computer and the brain. Yale University Press, 1958.
[2] W. Maass, “Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons,” Adv. Neural Inf. Process. Syst., vol. 9, pp. 211–217, 1997.
[3] W. Maass and C. M. Bishop, Pulsed Neural Networks, vol. 275. MIT Press, 1999.
[4] S. B. Furber et al., “Overview of the SpiNNaker system architecture,” IEEE Trans. Comput., vol. 62, no. 12, pp. 2454–2467, 2013.
[5] C. Johansson and A. Lansner, “Towards cortex sized artificial neural systems,” Neural Networks, vol. 20, no. 1, pp. 48–61, Jan. 2007.
[6] P. A. Merolla et al., “A million spiking-neuron integrated circuit with a scalable communication network and interface,” Science (80-.J.), vol. 345, no. 6197, pp. 668–673, Aug. 2014.
[7] B. V. J. Battisti et al., “Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations,” Proc. IEEE, vol. 102, no. 5, pp. 699–716, May 2014.
[8] J. Schemmel, D. Br"{u}derle, A. Gri"{o}bl, M. Hock, K. Meier, and S. Milbner, “A wafer-scale neuromorphic hardware system for large-scale neural modeling,” in Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2010, pp. 1947–1950.
[9] J.-S. Seo et al., “A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons,” in 2011 IEEE Custom Integrated Circuits Conference (CICC), 2011, pp. 1–4.
[10] Y. Kim, Y. Zhang, and P. Li, “A Reconfigurable Digital Neuromorphic Processor with Memristive Synaptic Crossbar for Cognitive Computing,” ACM J. Emerg. Technol. Comput. Syst., vol. 11, no. 4, pp. 1–25, 2015.
[11] G. Q. Bi and M. M. Poo, “Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type,” J. Neurosci., vol. 18, no. 24, pp. 10464–10472, 1998.
[12] F. Ponulak, “ReSuMe-new supervised learning method for Spiking

| Proposed synapse | Simultaneous read-write | Clock-less read-write | Additional remarks |
|------------------|------------------------|----------------------|--------------------|
| Y. Kim et al. [10] | Continuous, using PWM | 1R memristor | No | No |
| Serrano-Gotarredona et al. [26] | Continuous, using write-pulse superposition | 1R memristor | Yes | Yes |
| Wu et al. [20] | Continuous, using write-pulse superposition | 1R memristor | No | Yes |
| This work | Continuous, using write-pulse superposition | 1R memristor | Yes | Yes |

| Change in weight | Key features |
|------------------|--------------|
| Digital, multi-level | 8T-SRAM |
| Continuous, using PWM | 1R memristor |
| Continuous, using write-pulse superposition | 1R memristor |
| Continuous, using write-pulse superposition | 1R memristor |

| Additional remarks |
|--------------------|
| Multiple Clock phases are used for independent learning and recognition phases. |
| Additional inter-layer communication channel needed. |
| Reading disabled during write-pulse application |
| No support for recurrent networks |
Neural Networks,” *Inst. Control Inf. Eng. Pozn. Univ.*, vol. 22, no. 2, pp. 467–510, 2005.

[13] J. H. Lee, T. Delbruck, and M. Pfeiffer, “Training Deep Spiking Neural Networks using Backpropagation,” *Front. Neurosci.*, vol. 10, no. 508, pp. 1–10, 2016.

[14] E. Hunsberger and C. Eliasmith, “Spiking Deep Networks with LIF Neurons,” *arXiv:1510.08829 [cs]*, pp. 1–9, 2015.

[15] P. Sheridan, W. Ma, and W. Lu, “Pattern recognition with memristor networks,” *Proc. - IEEE Int. Symp. Circuits Syst.*, pp. 1078–1081, 2014.

[16] S. Park et al., “Nanoscale RRAM-based synaptic electronics: toward a neuromorphic computing device,” *Nanotechnology*, vol. 24, no. 38, p. 384009, 2013.

[17] J. M. Cruz-Albrecht, T. Derosier, and N. Srinivasa, “A scalable neural chip with synaptic electronics using CMOS integrated memristors,” *Nanotechnology*, vol. 24, no. 38, p. 384011, 2013.

[18] J. J. Yang and R. S. Williams, “Memristive devices in computing system,” *ACM J. Emerg. Technol. Comput. Syst.*, vol. 9, no. 2, pp. 1–20, 2013.

[19] B. Rajendran et al., “Specifications of nanoscale devices and circuits for neuromorphic computational systems,” *IEEE Trans. Electron Devices*, vol. 60, no. 1, pp. 246–253, 2013.

[20] X. Wu, V. Saxena, and K. Zhu, “Homogeneous spiking neuromorphic system for real-world pattern recognition,” *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 5, no. 2, pp. 254–266, 2015.

[21] Z. Wang, S. Ambrogio, S. Balatti, and D. Ielmini, “A 2-transistor/1-resistor artificial synapse capable of communication and stochastic learning in neuromorphic systems,” *Front. Neurosci.*, vol. 9, no. JAN, pp. 1–11, 2015.

[22] S. Ambrogio et al., “Neuromorphic Learning and Recognition With One-Transistor-One-Resistor Synapses and Bistable Metal Oxide RRAM,” vol. 63, no. 4, pp. 1508–1515, 2016.

[23] A. Shukla, V. Kumar, and U. Ganguly, “A software-equivalent SNN hardware using RRAM-array for asynchronous real-time learning,” *2017 Int. Jt. Conf. Neural Networks*, pp. 4657–4664, May 2017.

[24] A. Biswas, S. Prasad, and U. Ganguly, “A simple and efficient SNN and its performance and robustness evaluation method to enable hardware implementation,” 2016.

[25] P. Kumbhare et al., “A Comprehensive Study of Effect of Composition on Resistive Switching of HfAl1-xOy based RRAM devices by Combinatorial Sputtering,” *MRS Proc.*, vol. 1729, pp. 65–70, 2015.

[26] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, “STDP and sSTDP variations with memristors for spiking neuromorphic learning systems,” *Front. Neurosci.*, vol. 7, no. 7 FEB, pp. 1–15, 2013.

[27] G. Pedretti et al., “Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity,” no. May, pp. 1–10, 2017.