Neuromorphic Computing: From Devices to Integrated Circuits

V. Saxena[1, a)]
Electrical and Computer Engineering Department, University of Delaware, Newark, DE.

(Dated: 11 November 2020)

A variety of nonvolatile memory (NVM) devices including the resistive Random Access Memory (RRAM) are currently being investigated for implementing energy-efficient hardware for deep learning and artificial intelligence at the edge. RRAM devices are employed in the form of dense crosspoint or crossbar arrays. In order to exploit the high-density and low-power operation of these devices, circuit designers need to accommodate their non-ideal behavior and consider their impact on circuit design and algorithm performance. Hybrid integration of RRAMs with standard CMOS technology is spurring development of large-scale neuromorphic system-on-a-chip (NeuSoC). This review article provides an overview of neuromorphic integrated circuits using hybrid CMOS-RRAM integration with an emphasis on spiking neural networks (SNNs), device nonidealities, their associated circuit design challenges, and potential strategies for their mitigation. An overview of various SNN learning algorithms and their co-development with devices and circuits is discussed. Finally a comparison of NVM-based fully-integrated neuromorphic ICs is presented along with a discussion on their future evolution.

PACS numbers: 84.30.-r, 84.35.+i,
Keywords: Artificial Intelligence (AI), Back-end-of-the-line (BEOL) Processing, CMOS Neurons, Edge-AI, Non-Volatile Memory (NVM), Neuromorphic Computing, Resistive Random Access Memory (RRAM).

I. INTRODUCTION

Over the last decade, *deep neural networks (DNNs)*, or deep learning, have emerged as the next wave of Artificial Intelligence (AI) which has been propelled by the advances in specialized hardware, open-source software and availability of datasets. Hardware platforms such as the graphics processing units (GPUs) and application-specific integrated circuits (ASICs) for AI acceleration enable parallel processing of a large amount of data to train DNN models. Training or learning in DNN models is performed using variants of the gradient-descent *back-propagation, or Backprop*, algorithm[1–3] which is both computationally and energy intensive due to the massive amounts of data continuously shuttled between memory and processing units. Recently, low-power GPUs and ASICs have appeared for deep learning inference for Edge-AI[4,5]. However, training is performed on a GPU-based server or Cloud infrastructure using software frameworks such as TensorFlow[6] and PyTorch[7]. In view of diminishing returns from such architectures with the near-end of Moore's scaling[8], semiconductor industry's International Roadmap for Devices and Systems (IRDS) looks forward to *Beyond-Moore* or *post-CMOS* technologies to conceive radically new computing architectures for AI workloads[9]. This requires a cross-layer investigation of non von Neumann computing architectures across the entire devices, circuits and algorithms hierarchy.

Several classes of emerging non-volatile memory (NVM) devices are currently being investigated for their application in analog implementation of Edge-AI hardware. These devices can be two- or three-terminal and employ a wide range of material systems and associated physical mechanisms to achieve multi-level non-volatile memory states. Moreover, these NVM devices need to be monolithically integrated with standard CMOS technology to enable hybrid integrated circuit design. Deep learning hardware realizations include *vector-by-matrix multipliers* (VMMs), and neural-inspired or *Neuromorphic* computing circuits. The NVM devices are employed in the form of crossbar, or cross-point, arrays with or without selectors along with CMOS circuits at the periphery of the array.

While these emerging in-memory computing architectures show promise, in order to exploit the high-density and operation of these devices, *integrated circuit* (IC) designers need to accommodate their realistic behavior and nonidealities. This is particularly important for optimizing hybrid transistor-NVM circuit design for performance, area and power consumption. NVM nonidealities include device variability, low resistances offered by the two-terminal devices, resolution and stability of multi-level states, nonlinearity and device endurance. Moreover, novel in-situ learning algorithms must be developed which can take advantage of the localized in-memory computing to minimize von Neumann bottlenecks.

Several recent review articles in the literature focus on emerging devices for in-memory computing[10–16], neuromorphic learning algorithms[17], pathways and survey of neuromorphic hardware architectures[18,19]. This article provides the motivation and overview for spike-based Neuromorphic hardware that can be realized using emerging NVMs, with a specific focus on the Resistive Random Access Memory (RRAM), *aka* the *memristors*, and *spiking neural network* (SNN) algorithms. The unique aspect of this review is that it focuses on bridging

a) Electronic mail: vsaxena@udel.edu

the large gap between the expectations produced by the experimental results from emerging NVM devices and the requirements set by the neuromorphic system architectures. This is attempted by considering the device non-idealities that the circuit designers need to accommodate and the hardware bottlenecks to be surmounted when implementing spike-based deep learning in neuromorphic hardware. Rest of the article is organized as follows: Section II introduces SNNs and their use in neuromorphic computing. Section III provides an overview of the RRAM devices and discusses the impact of device non-idealities on circuit design considerations and system-level performance. Section IV discusses CMOS integrated circuits needed to realize neuromorphic computing. Section V provides a brief overview of the learning algorithms for deep SNNs. Finally, Section VI benchmarks the performance of fabricated NVM-based neuromorphic computing ICs followed by a concluding discussion.

II. SPIKING NEURAL NETWORKS (SNNS) AND NEUROMORPHIC COMPUTING

Energy-efficiency of DNNs realized on GPUs and ASICs based on von Neumann architectures is fundamentally limited by the energy and latency cost of the 'distance' between storage memory and processing units[20]. In a radical contrast, a biological brain stores memory and performs localized computing using similar neural motifs with extremely high energy-efficiency, thus making a compelling inspiration for in-memory computing for DNNs.

Fig. 1 illustrates a neuromorphic computing architecture based on 1T1R crosspoint arrays with neuron circuits at the array periphery. In a fully-connected neural network, each array realizes a network layer which are connected with each other using on-chip and/or off-chip interconnects. These interconnects employ asynchronous spike-based communication using a protocol such as the address-even representation (AER) protocol, which is widely used in neuromorphic sensors and processors[21–23]. The neural network weights are stored in the NVM memory array. The pre-neurons concurrently drive the rows (or the wordlines), as opposed to random access. The pre-neuron activations (or voltages) are weighted by the conductance of the synaptic weights and the resulting current is summed and integrated on the post-neurons connected to the columns (or the bitlines).

In the past decade, advances in spike-based models with localized plasticity mechanisms such as the *spike-timing-dependent-plasticity (STDP)* and its feedback-based modulation[24–30] have opened new avenues in neuromorphic computing research. For example, theoretical studies have suggested STDP-like plasticity mechanisms can be used to train two-layer SNNs in-situ without trading-off their parallelism[31–34]. SNNs essentially encode information using asynchronous spatiotemporal 'spikes.' A spikes sequence can represent the input signal either using *rate-coding* where the average spike rate represents a real valued signal, or temporal coding where spike delay (or latency) encodes the same information. In biology, sensory information is encoded as a combination of both rate and latency coding.

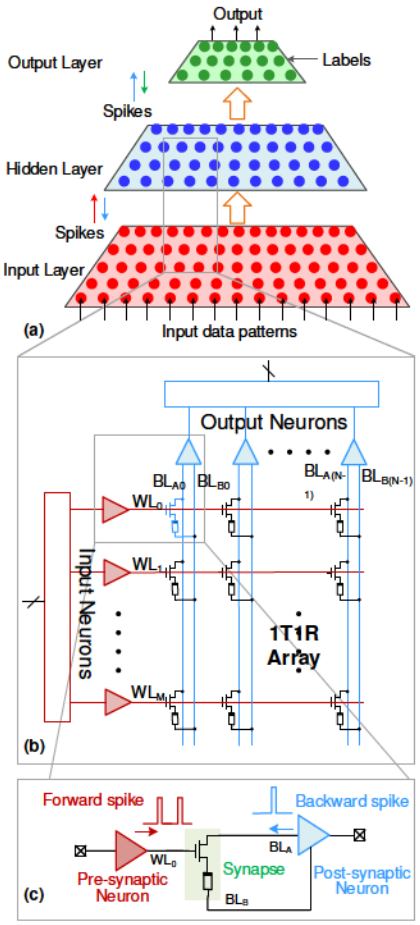

FIG. 1. A neuromorphic computing architecture built around 1T1R RRAM crosspoint arrays: (a) A fully-connected spiking neural network (SNN) showing input, hidden and output layers of spiking neurons, (b) a crosspoint RRAM array with input and output neurons, (c) a single synapse between the input and output neurons with localized weight updates.

A. Leaky Integrate and Fire Neurons

Biological neurons exhibit complex spike signal processing and spike generation behavior with some of the spike filtering occurring in the dendrites[35]. As a result, wide range of models have been developed in the literature to emulate their response[21,36–40]. However, in neuromorphic computing only the simple salient features useful for learning and inference are adapted into the SNN architecture. A single-compartment *leaky integrate-and-fire* (LIF) neuron is a simple and commonly-used neural motif. Network weights, w_{ij}, are stored in 'synapses' which

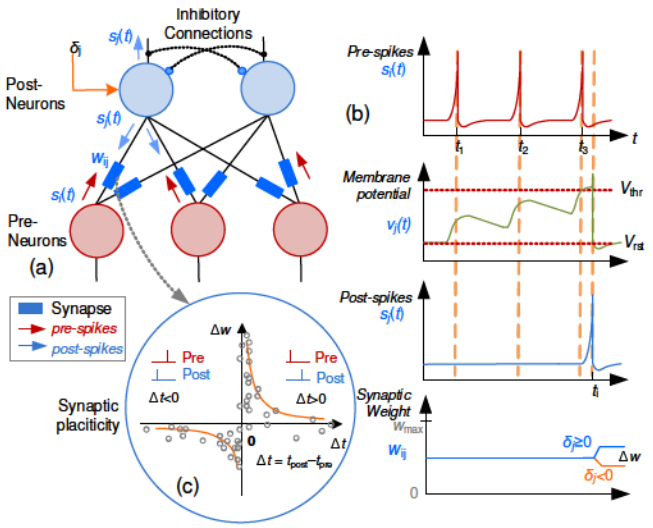

FIG. 2. (a) Section of a two-layer spiking neural network illustrating pre and post spike flow. (b) Transient waveforms for these spikes, membrane potential, and weight update modulated by an external feedback error, δ. (c) Spike-timing dependent plasticity (STDP) learning window where the weight update depends upon the temporal delay, Δt, between the pre and post spikes at a synapse.

can be updated based on Hebbian learning, i.e. the correlation of the spiking activity of the **pre**-synaptic and **post**-synaptic neurons. During feedforward passes, a LIF neuron integrates its input spikes, $s_i(t)$, weighted by the synapses, into its membrane potential

$$v_{m,j}(t) = \sum_{i=1}^{N} w_{ij} \int_{-\infty}^{t} s_i(t) \otimes h(t) dt \qquad (1)$$

where $h(t) = e^{-t/\tau_m} u(t)$ is the impulse response that incorporates neuron's leaky behavior. When the membrane potential of a neuron crosses its firing threshold, i.e. $v_{m,j}(t) > V_{thr,j}$, the neuron produces a post-spike, $s_j(t)$. A firing event resets the membrane potential to a resting potential, V_{rst} and propagates the post-spike in the forward as well as the backward direction. This is described by Eq. 2 below and illustrated in Fig. 2.

$$\text{at } t = t_j^f \quad v_{m,j}^l(t) > V_{thr,j}^l : \begin{cases} v_{m,j}^l(t) \leftarrow V_{rst} \\ s_j(t) \leftarrow g(t - t_j^f) \end{cases} \qquad (2)$$

After firing, a neuron enters a refractory, or silent, period before it can fire again. During the refractory period of duration τ_{refr}, membrane potential rests at V_{rst}. τ_{refr} sets the maximum rate at which a neuron can fire, i.e. $\nu_{max} \geq \frac{1}{\tau_{refr}}$, and thus the resolution for quantization in the time-domain.

Neurons can form inhibitory and/or excitatory connections with other neurons as seen in Fig. 2. Inhibitory connections prevent other neurons from firing if one of the neurons has fired in a given time window. Excitatory connections make connected neurons fire simultaneously. Inhibition is used to implement competitive learning in an population of neurons to realize Winner-take-all (WTA) motifs[41].

B. Synapses and Plasticity

Synapses essentially implement weights in the SNN where the pre-spikes are weighted by the conductance of the synapse before they are integrated in the neuron. In case of online learning, synaptic weights are in-situ updated on the neuromorphic chip. Weight update in a synapse is governed by a plasticity rule based on Hebbian correlation which can be modulated by an error feedback, leading to an approximated three-factor learning rule[42]

$$\Delta w_{ij} = f(t_i, t_j, \delta_j) \qquad (3)$$

Here, t_i and t_j are pre and post spike times, and δ_j is the error feedback. Several plasticity rules have appeared in literature based on electrical probing of biological neurons and update rule derivations in computational neuroscience[26,29,43]. STDP rule is a two-factor rule based on the time difference between post and pre spikes $\Delta t_{ij} = t_{post} - t_{pre} = t_j - t_i$ and expressed as

$$\Delta w_{ij} = \begin{cases} a^+ e^{\frac{\Delta t_{ij}}{\tau^+}}, & \Delta t_{ij} = t_j - t_i \geq 0 \\ -a^- e^{-\frac{\Delta t_{ij}}{\tau^-}}, & \Delta t_{ij} = t_j - t_i < 0 \end{cases} \qquad (4)$$

Here, a^+ and τ^+ are the voltage and temporal parameters for the *long-term potentiation* (LTP) when the synaptic weight increases. Conversely, Here, a^- and τ^- are the corresponding parameters for the *long-term depression* (LTD) when the synaptic weight decreases.

Fundamentally, spike-based computing simplifies the need for precise multiplication and replaces it by scaling of bi-level, or binary, spikes by synaptic conductances. This results in simpler digital neuromorphic hardware implementations and precludes the need for four-quadrant multiplication in analog neuromorphic realizations. Consequently, SNNs can perform computing with simpler hardware and consume very small amount of energy only when the spike events occur[23,44,45]. Due to their unique characteristics, SNNs are better realized on an event-driven neuromorphic platform instead of a von Neumann computer.

C. Analog Mixed-Signal Neuromorphic Platforms

As discussed earlier, neural-inspiration provides the motivation for realizing dense and low-power neuromorphic hardware by implementing neuron and synapses using analog mixed-signal circuits. Advances in analog neuromorphic circuits include the Neurogrid hardware[46],

where subthreshold biomimetic CMOS circuits were developed to reproduce dynamics occurring in biological neural networks. However, the fundamental limitation of such analog architectures is that the weights are dynamically stored and updated on capacitors, which leak away in a few seconds, thus limiting any long-term learning. Bistability of analog weights has been used as a stop-gap solution[47,48], but recent studies on deep neural networks have determined that 4-bit or higher synaptic resolution is needed to realize SNNs with reasonable classification accuracy[49,50]. Other solutions include using *Floating Gate, or NOR Flash*, devices for realizing multilevel non-volatile synaptic weights[51]. However, despite of their excellent retention, floating-gate memory suffers from low endurance of $< 10^5$ write cycles. This limits the number of times the neural network weights can be updated and thus significantly limit on-chip training capability.

D. Neuromorphic Computing using Emerging Nonvolatile Memory Devices

Several emerging *nonvolatile memory* (NVM) devices including RRAMs, *spin-torque transfer RAM* (STTRAM), *phase-change RAM* (PCRAM), also generally referred to as memristors, have been considered for their use in Edge-AI and neuromorphic computing[20,55–59]. Recently, ferroelectric-field-effect transistors (FeFETs) have come under focus for their low power performance[54,60]. In the last decade, these NVM devices have been extensively investigated as a high-density potential replacement for Flash memory[20,52,61] with their advantages summarized in Table I. As the research community gain a better understanding of the physical mechanisms for electrical switching in the RRAM devices, there is a trend towards developing novel applications by leveraging these devices for computing[57,62–67].

These emerging NVM devices are employed in the form of *crossbar or cross-point arrays* with a diode or transistor selector— cell with *one transistor and one RRAM* (1T1R). These NVM arrays are being investigated for DNN computations in the analog domain. These include VMM for Edge-AI[58,68] and synaptic weights in a mixed-signal neuromorphic hardware[23,69]. Emerging NVM arrays are promising for neuromorphic computing as they provide: (1) a very high synaptic density with low leakage power, (2) localized in-memory learning similar to biological synaptic plasticity, (3) very low power consumption with event-driven updates, and (4) non-volatility of weights[23]. This review focuses on two-terminal RRAM devices for neuromorphic computation.

III. RRAM DEVICE CHARACTERISTICS AND PROCESS INTEGRATION

A. RRAM Devices

Resistance switching in emerging nanoscale resistive memory devices has sustained interest with the goal of high-density and lower power replacement for NVM-based embedded memory and computing applications[9,10,52,70]. RRAM arrays are considered as a suitable alternative to the Flash-based solutions due to the following reasons:

(i) The filamentary nature of resistance switching has the potential to scale well beyond the sub-10nm feature size[61,71].
(ii) Lower switching voltages allow low power operation and compatibility with scaled-CMOS[952].
(iii) Very simple planar two-terminal structures and fabrication-friendly materials facilitate integration with standard CMOS technology.
(iv) These devices demonstrate biologically plausible plasticity (i.e. weight update) behavior in several experiments,[57,62,72–74] and therefore have emerged as an ideal candidate for realizing electrical equivalent of biological synapses.

Several categories of RRAMs have been intensely pursued by the device community: (1) Electrochemical memory (ECM) *aka* Conductive Bridge RAM (CBRAM), (2) Valence Change Memory (VCM) *aka* Oxygen Vacancy based RAM (OxRAM), (3) Thermochemical memory (TCM), and (4) Interfacial or 2D switching RRAMs[75–77]. This review primarily focuses on CBRAM and OxRAM devices for neuromorphic computing, which are graphically depicted in Fig. 3. All these RRAMs are essentially metal-insulator-metal (MIM) structures where the resistance between the two electrodes can be changed in a non-volatile manner by either filamentary or interfacial-type switching.

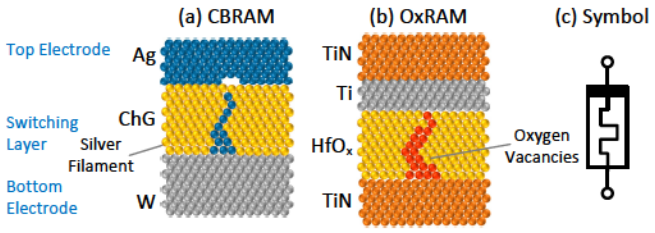

FIG. 3. Graphical illustration of the cross-section of CBRAM and OxRAM NVM devices and their circuit symbol. Device cross-sections are not to the scale.

CBRAM: CBRAM uses a dielectric layer sandwiched between two electrodes and the resistance is changed by forming an conductive electrochemical bridge. As shown in Fig. 3(a), the anode (top electrode) is electrochemically active and is made of metals such as Silver (Ag)[78–81] or Copper-based alloys (e.g. $CuTe_x$)[82].

TABLE I. Comparison of memory technologies for in-memory computing. Data reproduced from Refs.[20,52–54].

Parameters	SRAM	DRAM	NOR Flash	PCRAM	STTRAM	OxRAM	CBRAM	FeFET
Cell size	$100F^2$	$7F^2$	$5F^2$	$4F^2$	$12F^2$	$4\text{-}6F^2$	$4\text{-}6F^2$	$24F^2$
Density	Low	High	High	V. High	High	V. High	V. High	High
Write Latency	1ns	5ns	10μ-1ms	100ns	2-25ns	10ns	10ns	3ns
Read Latency	1ns	20-80ns	50ns	10ns	2-25ns	1-10ns	10ns	2ns
Write Energy (pJ/bit)	<1	<1	100	2-25	0.1-2.5	0.1-3	0.1-10	0.1
Leakage Power	High	Medium	Low	Low	Low	Low	Low	Low
Endurance (write cycles)	$> 10^{15}$	$> 10^{15}$	10^5	10^8	10^{12}	10^9	10^8	$> 10^5$
Switching Polarity	NA	NA	Bipolar	Unipolar	Bipolar	Bipolar	Bipolar	Bipolar
MLC Capability	✗	✗	4-8 bits	4-6 bits	2 bits	4-6 bits	2 bits	3 bits
MLC Retention	✗	✗	Years	R-drift	Tunneling	R-drift	R-drift	Tunneling
3D Stacking	✗	✗	✗	✓	✗	✓	✓	✓

The cathode is an inert electrode and usually realized using Platinum (Pt), Nickel (Ni), or Tungsten (W). The dielectric is a thin film layer which comprises of either a chalcogenide (such as Ge_xSe_{1-x}), amorphous silicon[71,72], Al_2O_3, MO_x, or HfO_x[82]. The reader is referred to[81,83] for the CBRAM fabrication details. An initial step called 'forming' step is required to introduce silver into the unformed or pristine switching layer. This is achieved by either applying a sufficiently large voltage pulse (*aka* electroforming) or by photo-diffusion using optical illumination[84,85].

OxRAM: OxRAMs are fabricated as an MIM stack with transition metal oxides as the dielectric. The mechanism for resistance switching is the formation of a conductive filament due to the migration of oxygen vacancy defects[86,87]. Several insulators have been explored in the literature including HfO_x, Ta_2O_x, TiO_x, ZrO_x, and NiO. Since forming an oxygen vacancy in these oxides requires higher energy, a metallic layer of Hf (or Ti, La, Zr) is used as a *scavenger electrode* that scavenges oxygen vacancies from HfO_x[88]. This stack is shown in Fig. 3(b). The Hf/HfO_x (or Ti/HfO_x, etc.) interface acts as the active electrode[89] which is capped with material such as TiN, Ni, TaN, ITO, or Al[87]. Incorporation of the scavenger electrode leads to better switching control, reduced variability and higher retention[88]. Doping of OxRAM active layer with dopants such as Ti or Ge increases the oxygen vacancies and has been explored to realize forming-free devices[87].

Other RRAM Variants: Example of TCM devices include transition metal oxide cells such as NiO and HfO, where filamentary switching occurs due to current-based heating and the resulting stochiometric changes due to temperature increase[75,77,90]. Interfacial switching RRAMs include Nb-doped $SrTiO_3$ (NbO) and doped perovskites such as $Pr_{0.7}Ca_{0.3}MnO_3$ (PCMO). These RRAMs are also referred to as 2D switching devices as the resistance is dependent on cross-section area, as opposed to the filamentary switching devices[75,91,92].

B. RRAM Electrical Characteristics

CBRAM: A typical hysteresis current–voltage (I–V) characteristics in a metal–insulator–metal (MIM) structure are shown in Fig. 4. These electrical switching characteristics were obtained from experimental characterization of $Ag/Ge_{20}Se_{80}/W$ CBRAM devices which were fabricated by Mitkova group at Boise State University[81,83]. Here, triangular voltage sweeps were applied across the device and the current was measured using a Semiconductor Parameter Analyzer (SPA) such as Keysight B1500. The I-V sweeps were performed for several settings of the compliance current, I_{cc}, ranging from 50nA to 10μA. The device state during the I-V switching characteristics are described as follows:

(A) Here, initially the device is in the *High-resistance state* (HRS), erased, or Off state.

(B) When a sufficiently large positive voltage greater than the program threshold voltage (V_{th}^+) is applied on the top electrode, silver is oxidized and the Ag^+ ions start moving towards the cathode and forming a bridge (or filament) in the process.

(C & D) The silver ion bridge eventually forms a high-conductivity path between the electrodes realizing the *Low-resistance state* (LRS), programmed, or On state. This is referred to as the Program or Set operation.

(E) Conversely, if a voltage more negative than the erase threshold voltage (V_{th}^-) is applied across the CBRAM, the filament is dissolved and the device reverts to the HRS. This is called the Erase or Reset operation.

The electrochemical process of conductive filament formation is inherently stochastic and varies across devices and switching cycles, where silver filaments of varying geometry can be formed in the amorphous switching layer. The variability is reduced with scaling to sub-15nm cross-section area as only one dominant filament can be formed.

CBRAM switching thresholds depend upon the active electrode material and switching layer used in the device. In the CBRAM fabricated by author's collaborators with their experimental switching characteristics shown in Fig. 4, the program threshold is $V_{th}^+ = 0.7V$ and the

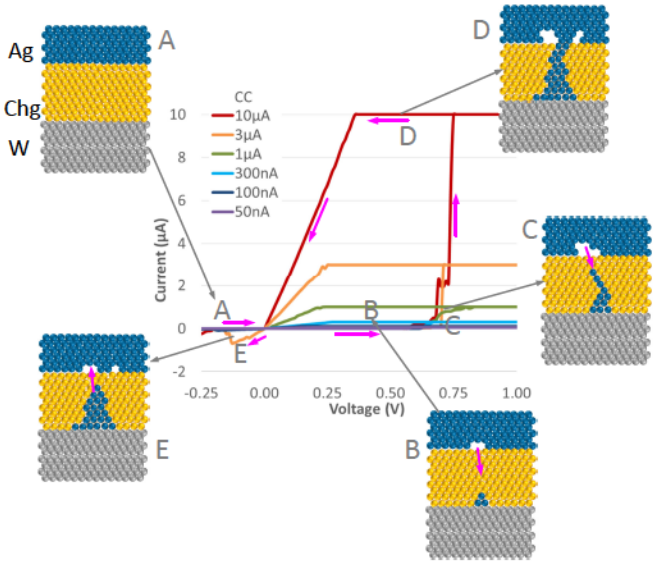

FIG. 4. Experimental I-V characteristics of a CBRAM device with graphical illustration of Program (A-D) and Erase (E) operations.

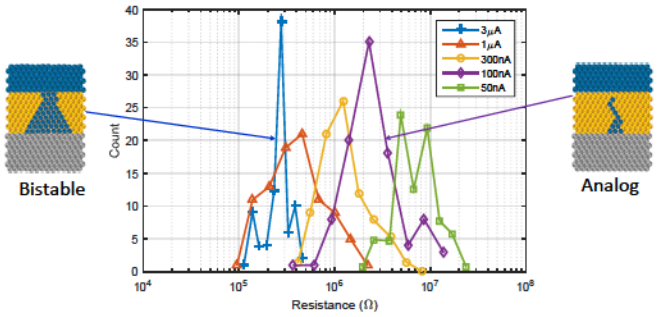

FIG. 5. CBRAM resistance distribution for several values of compliance current. Analog-like states are observed when a weak filament is formed. A thick filament leads to digital switching behavior.

erase threshold is $V_{th}^{-} = -0.1V$. The CBRAM switching behavior is bipolar, i.e. the Set and Reset operations require different polarity. CBRAM' bipolar switching is asymmetric as a very small negative voltage is needed to break the filament (bridge) at the anode and revert the device to HRS[81,93,94].

Compliance current is an important parameter that sets the maximum current in the device during the electroforming, and the subsequent operation of the device, and thus essentially determines the LRS resistance, R_{LRS}. This is illustrated in Fig. 5, where the resistance distribution for varied values of I_{cc} is plotted. For a large $I_{cc} > 10\mu A$, a thick filament is formed resulting in a narrower distribution with lower R_{LRS} values. This regime is suitable for digital applications with binary or bistable memory.

For lower $I_{cc} \sim 1\mu A$, a weak filament is formed with a wider distribution and higher resistance values. These characteristics give rise to the possibility of realizing analog-like behavior where the RRAM resistance can be programmed to one of the multilevel resistance states. This can be thought of as a *multi-level cell* (MLC) with 2-bit or higher resolution. The filament structure determines the switching speed, repeatability, LRS resistance range, multilevel behavior, and the retention of devices.

OxRAM: Electrical characteristics of OxRAM follow a I-V hysteresis loop similar to CBRAM with a few differences. Resistance switching in Ti/HfO_x (Hf/HfO_x,Ta/Ta_2O_x and similar stacks) OxRAMs occurs due to the formation of a conductive filament of oxygen vacancies. The metallic Hf, Ti or similar layers, facilitate scavenging of oxygen atoms from HfO_x layer. The energy of breaking oxygen (O) is compensated by the exothermic Hf-O bond formation energy[88,89]. When a positive voltage is applied to the active electrode, positive oxygen vacancies (V_O^+) are created at the Ti/HfO_x interface, which acts as a V_O^+ reservoir. The positively charged vacancies diffuse towards the inert (TiN) electrode held at a lower potential, thus forming a conductive filament. As the filament grows in length, the electron conduction mechanism changes from trap-assisted tunneling to Poole-Frenkel hopping to eventually Ohmic conduction when the filament is fully formed and consigns the OxRAM to LRS[89]. A compliance current ensures that complete breakdown of HfO_x dielectric is avoided. A 1T1R cell configuration is typically used to set this compliance current in the array. When the voltage polarity across the electrodes is reversed, O^- diffuses back into HfO_x and the conductive filament is dissolved leading to HRS.

OxRAMs devices based on transition metal switching layer have been extensively characterized using 1R as well as IT1R arrays[82,88,95–99]. The program and erase thresholds vary widely depending upon specific device structure and material used. For example in devices from Leti[95,100], the Program/Set thresholds in the range of $V_{th}^{+} \approx 1V$ and Erase/Reset is $V_{th}^{-} \approx -1.5V$. The bipolar switching in OxRAM is more symmetric than CBRAMs as a larger negative voltage is needed to diffuse the O^- ions back and dissolve the V_O^+ bridge. The LRS variability depends on I_{cc} and the pulse duration during the initial forming step. The HRS variability depends upon the target resistance range[95], which is of interest for analog (or multilevel) applications. For HRS, cycle-to-cycle variability tends to higher than the cell-to-cell variability. Thus a smart program-and-verify algorithm is needed to place the multi-level OxRAM in a specific resistance state. The forming voltage, ≈2V, is larger than the program threshold. A smaller $I_{cc} \approx 5\mu A$ and shorter pulse width (≈ 100ns) avoids stuck-at-LRS faults and places the OxRAM in analog region of operation (10kΩ-1MΩ). The program/Set operation can be abrupt in OxRAMs due to a positive feedback loop between electrical field/temperature and oxygen vacancies. The ana-

log behavior can be improved by introducing a *thermal enhanced layer* (TEL) to provide a better control over incremental resistance changes[101?] .

Other RRAMs: TCM devices exhibit apolar or unipolar switching, i.e. the Set and Reset operations have the same polarity[75,90]. The advantage of unipolar switching is that a simple diode-based selector can be employed forming a 1D1R cell. Disadvantages of TCM devices include the higher energy consumption in the thermal-activated switching process[90]. Interfacial switching devices such as NbO and PCMO exhibit bipolar switching by modulating the Schottky barrier barrier at the metal-oxide interface[75,91,92]. An advantage of the interfacial switching RRAMs is that they exhibit an asymmetric nonlinearity, which realizes a selector-free self-rectifying cell. Also, forming-free PCMO RRAM devices have been demonstrated which is advantageous for on-chip integration[102].

C. RRAM Retention and Endurance

Retention and endurance of both CBRAM and OxRAM have been extensively studied for binary storage applications[52,53]. Development of RRAMs for analog computing where each cell can be programmed with multilevel states is a current focus of research[103–107]. Ti/HfO_x OxRAMs have demonstrated endurance of $\approx 10^9$ cycles, ultra-fast writes ($\sim$ 5ns), and R_{off}/R_{on} ratio exceeding 100[52,88,98]. Multilevel resistance states are programmed by modulating either the voltage amplitude or the pulse-width of the program/erase pulses, and setting a compliance current. A program-and-verify scheme is employed for iterative programming to achieving tight resistance distributions while minimizing the read/write disturb between the cells[108]. Consequently, the device is stressed multiple times per MLC write event and has a bearing on overall device lifetime. Recent work on MLC OxRAM characterization has shown that it can be programmed up to 64-levels, or 6-bit per cell resolution[66,108–110]. However, the programmed states have been observed to relax and their resistances drift over time on a timescale of hours to weeks[66,108,111]. Several retention failure modes have been observed depending upon the device material stack and structure[112]. Also, the resistance drift is accelerated as the chip temperature is increased due to the increased Brownian motion of conducting ions in the switching layer[66].

CBRAMs have also shown multi-level resistance states with 2-bit/cell storage and $\geq 10^5$s retention by using appropriate compliance current settings[53,113]. Based on literature survey, OxRAMs have been more amenable to analog behavior when compared to CBRAMs, but the latter can improved by using optimized bilayer materials[53].

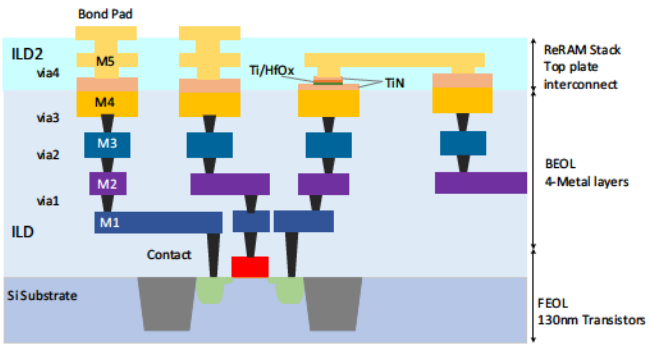

FIG. 6. Graphical illustration of integration of RRAM stack in the BEOL of a 4-Metal CMOS process as described in Ref.[100]

D. Process Integration of RRAM with CMOS

Integration of emerging NVM devices with standard CMOS technology is essential for the development of large scale neuromorphic circuits. To allow flexibility when optimizing the NVM device stack, a preferred approach is to integrate RRAM, or any other emerging NVM devices, in the back-end-of-the-line (BEOL) of a CMOS process. OxRAMs allow straightforward integration with a CMOS process flow as the materials used are compatible with CMOS fabrication. This is due to the fact that HfO_x and similar materials are used as high-k gate oxide in scaled CMOS process nodes. Recently, few foundries have integrated OxRAM and CBRAMs into their CMOS fabrication flows[20,69,114,115]. Fig. 6 shows a cross-section representation of the OxRAM integrated in the BEOL of a 4-metal CMOS process[100]. 1T1R RRAM arrays from 16Kbits to 2Mbits have been fabricated using this platform with memory size scaled down to 40nm[67,96]. CBRAM with Ag filament is challenging as silver is a contaminant for silicon-based devices. Thus, copper (Cu) based CBRAMs have been in focus as Cu is already used in the CMOS BEOL flow[53].

E. Challenges with RRAM Devices for Neuromorphic Computing

Hybrid CMOS-NVM circuits have been proposed to achieve dense integration of CMOS neurons and emerging memory for neuromorphic computing ICs[116–121]. However, RRAM synapses incur several limitations which are summarized as follows[23]:

1. **Variability:** Individual RRAM characteristics including the switching threshold voltages and intermediate resistances vary from cell-to-cell and cycle-to-cycle depending upon the resistance state and the compliance current[53,66,82,95,112]. Impact of variability on STDP updates was experimentally studied on a crossbar array[122]. The cell-to-cell

variation of switching thresholds is a major challenge that needs to be addressed by careful program/erase pulse waveform design[122]. In an 1T1R cell, transistor variations also contribute to variations in the overall synapse. The RRAM resistance also exhibits temperature-dependent switching with a decrease in R_{HRS} due to the increase in leakage current with temperature[107,123].

2. **Low-resistance**: Typical RRAMs exhibit 1-10$k\Omega$ LRS resistance. A large synapse fan-out incurs significant static power consumption in the CMOS driver circuits. A 1T1R synapse can mitigate this by employing a source-degenerated transconductor[124] or cascode topology[124,125].
3. **Resolution, Retention and State Drift:** Even though RRAM can realize multi-level cell (MLC) storage[66], the resistance-drift over time presents challenges in their use as stable analog synapses over a long duration. Furthermore, the resistance drift has strong dependence on the chip temperature which can make it difficult to estimate degradation of the synaptic states beyond acceptable performance limits.
4. **Endurance:** Continual on-chip training is restricted by the maximum number of write cycles allowed before the devices wear out. Also, MLC devices with shorter retention time will lead to neural network classification performance degradation if states are not periodically restored.

IV. NEUROMORPHIC INTEGRATED CIRCUITS

As seen earlier in Fig. 1, RRAM based neural networks are built around 1R or 1T1R crosspoint arrays driven by pre-neurons. The post-neurons sum the weighted currents and produce spike patterns. The neurons are located on the periphery where pre-neurons drive the rows (or the wordlines) and post-neurons on the columns (or the bitlines) accumulate weighted synaptic currents and generate output spikes.

A. CMOS-RRAM Synapses

An 1R synapse is simply a crosspoint synapse without a select transistor which is programmed to a resistance state of R_M[126,127]. The LIF neuron provides a virtual ground where the input spike voltages are weighted by the conductance, $G_M = \frac{1}{R_M}$, of the synapse and then integrated on the neuron. On the other hand, a transconductor synapse is realized with an 1T1R array where the select transistor is biased appropriately to set the synapse transconductance[124,128]. Fig. 7 shows a 1T1R synapse circuit which employs the transistor as a source-degenerated transconductor (G_{syn}) and converts the presynaptic spikes to synaptic currents (i_{spk}). Here, the pre-neuron output drives the input capacitance of the transistor ($\approx$ 5-10fF in 65nm CMOS). The bottom (inert) electrode is connected to V_{post} terminal, which stays at ground during integration and switches to V_{rst} when the post-neuron fires.

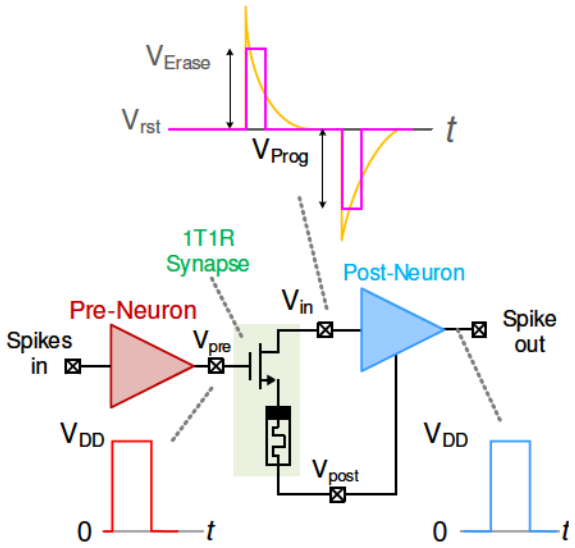

FIG. 7. A transconductance-type 1T1R synapse with in-situ weight updates as described in Ref.[124].

The synaptic current in Fig. 7 is expressed as

$$i_{spk}(t) \approx G_{syn} \cdot v_{pre}(t) \tag{5}$$

where the transconductance of the overall synapse is given by

$$G_{syn} = \frac{g_m G_M}{g_m + G_M} \approx g_m || G_M \tag{6}$$

Here, the transistor is biased in saturation with small-signal transconductance, g_m, and output resistance, r_o.

The relationship between i_{spk} and v_{pre} for the 1T1R synapse is nonlinear. This nonlinearity shown in Fig. 8 and depends upon the RRAM state and also vary with process and temperature. The resulting synapse transconductance, G_{syn}, is also not constant and dependent upon v_{pre}. By employing binary spikes, the I-V nonlinearity is mitigated as a straight line can be fitted across the two points on the I-V characteristics with a constant slope. Also, the spacing between various synaptic states can be made linear by controlled programming of the MLC states, or by accommodating the resulting nonlinearity at the algorithmic level.

The synapse allows weight updates based on the correlation of pre-synaptic and post-synaptic spikes and error feedback (δ) as illustrated in Fig. 9. Here, voltage *waveform engineering* is performed to translate the time delay between the pre- and post-spikes (Δt) into the application of program and erase pulse across the RRAM device to increase or decrease the weight, i.e. the

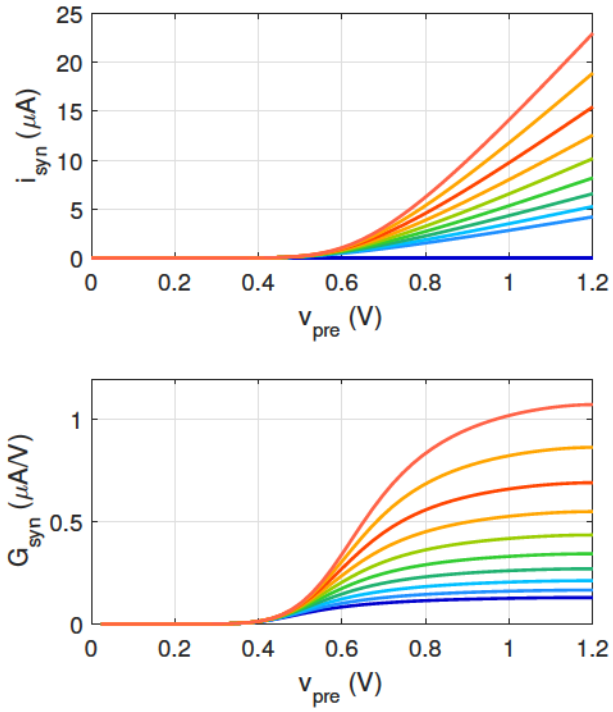

FIG. 8. (left) Simulated DC i_{syn} versus v_{pre} characteristics for a 1T1R synapse with 65nm CMOS and an RRAM model. The RRAM is initialized in the model in ten distinct states. The select transistor is sized with $\frac{W}{L} = \frac{600nm}{60nm}$ with its drain at $V_{rst} = 0.6V$ and source at $0V$. (right) Synapse transconductance, G_{syn}, as a function of input voltage.

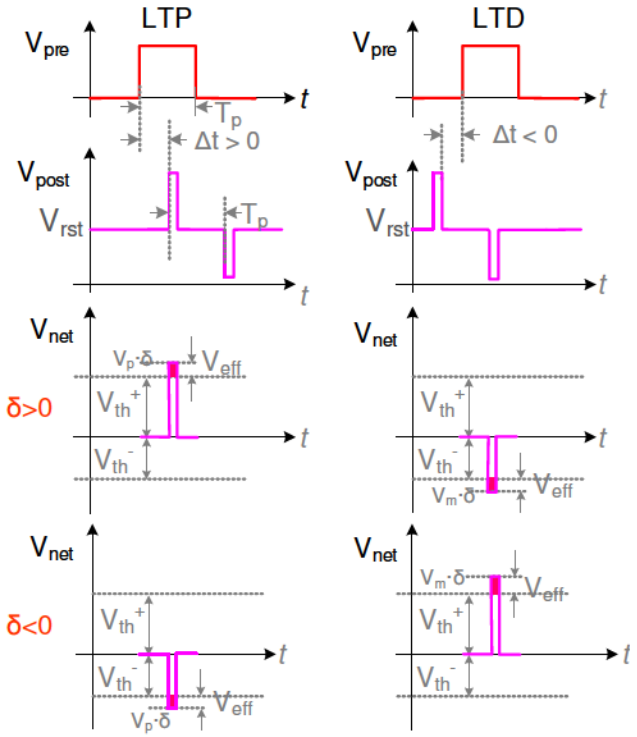

FIG. 9. An example of waveform engineering in the 1T1R synapse to translate the temporal correlation of spikes into synapse weight update as described in Refs.[124,128]. Rectangular as well as exponentially decaying pulse can be used to customize the STDP learning function, although only the former is shown here.

synapse transconductance, $\Delta w = \Delta G_{syn}$[128]. The pre-neuron only observes the input capacitance of the transistor and can drive a large synaptic fanout using digital CMOS buffers (as opposed to driving several resistances in parallel as in the 1R array). The output resistance of the transconductor synapse is given by

$$R_{out} \approx g_m r_o \cdot R_M \tag{7}$$

which significantly reduces the loading at the input terminal, V_{in}, of the post-neuron in the firing phase. When pre and post pulses overlap, the V_{post} node is loaded by

$$R_{post} \approx R_M + r_o \tag{8}$$

The select transistor should be sized such that during inference, the voltage dropped across R_M shouldn't exceed the threshold, V_{th}^+ to avoid disturbing the synaptic weight. Also, during the weight-update phase, a large fraction of the program/erase voltage should drop across R_M affect LTP/LTD. These constraints need to work in concert with the waveform design seen in Figure 9. Significant post-neuron loading and the associated energy consumption is avoided by ensuring sparse overlap of pre and post spikes at the algorithm level.

The pre and post waveforms are designed such that during LTP, a positive voltage pulse greater than the V_{th}^+ appears across the device. Conversely, during LTD, a voltage pulse more negative than V_{th}^- is applied across the device. An error feedback signal, δ, can be employed to modify the post-waveform by changing its sign and amplitude in order to implement a desired three-factor learning rule as in Eq. 3. In addition to waveform engineering with RRAMs, novel devices have been engineered with bio-realistic internal dynamics to support STDP-type updates. These include Ag-in-oxide memristor (Ag/SiOxNy/Pt) with diffusive dynamics[129] and second-order oxygen vacancy dynamics in a Pt/WO$_x$/W device[130].

B. Array Programming Circuits

The immediate application of NVM-based NeuSoCs is in transfer learning for inference applications. Also, on-chip learning requires suitable initialization of synaptic weights. Moreover, it may be desirable to read out learned weights from a particular chip. Thus, RRAM array program and read circuits become a necessary feature (in addition to the neurons).

For SNNs with unsigned weights a 1T1R synapse is employed. However, for VMMs and higher accuracy DNNs, signed weights become necessary. These are realized using a differential 2T2R topology where the conductances of two devices are subtracted in the analog domain[131–133]

$$G_{syn,ij} = G_{ij}^{+} - G_{ij}^{-} \tag{9}$$

Here, $G_{ij}^{\pm}$ are the conductances of positive and negative branches.

In order to program multi-level states in 1T1R (or 2T2R) arrays, an *Incremental Step Pulse Program* (ISPP) scheme is employed, which was adapted from earlier MLC Flash memory designs[134]. ISPP for RRAM arrays has been demonstrated in the recent literature using a semiconductor parameter analyzer (SPA). In these experiments, the voltage pulse profiles are set after careful characterization of cell-to-cell and cycle-to-cycle variability of the devices[135–137].

In an ISPP scheme, the program (or erase) voltage is applied to the device, while setting a compliance current, I_{cc}, and its state is verified using a read operation. The program (erase) pulse magnitude is gradually increased and the state is verified till a target synapse conductance state is achieved[135,136]. Alternatively, pulse width can also be progressively increased (or decreased) to set the desired conductance state[138], however, modification of the program pulse magnitude was found to be more effective in terms of device variability and program/erase time[136].

On-chip ISPP implementations include 1D arrays of row DACs and column ADCs with addressing logic and an on-chip RISC processor to isolate a particular MLC cell and then program it into a desired state[139]. Here, the DACs drive the worldlines with analog inputs, x_i, and the each of the columns digitize the weighted sums, $y_j = \sum_i w_{ij} x_i$. Backend processing is handled in the digital domain using the processor.

C. CMOS Neuron Circuit Design

There is a significant body of work over the past several decades on low-power spiking neurons and synapses using subthreshold CMOS circuits, which is documented in [21] and references therein. These circuits operated in the range of kHz spike rate, similar to biology, and were optimized for ultra low power emulation of neurophysiological behavior. However, with the advent of NVM synapses, a renewed focus is on in-memory neuromorphic computing for deep neural networks. Recently, several spiking neurons designs have been proposed for neuromorphic computing[36,40,140,142,145], however the challenges associated with their circuit integration with practical RRAM devices weren't considered.

In the context of NVM-based neuromorphic computing, CMOS neuron design can vary widely depending upon the desired functionality. For inference-only NeuSoC architectures, the neuron needs to integrate synaptic currents, make decisions when the membrane potential crosses the threshold (i.e. fire), and generate post spikes. On the other hand, for in-situ learning functionality, the neuron has to generate the waveform seen in Figs. 7 and 9 and drive the RRAMs (Eq. 7). Moreover, the neuron should also allow for a three-factor learning rule that incorporates the feedback error (δ).

Ultra-low power neurons with ≈100nW static power consumption can be designed for inference-only neuromorphic ICs[146]. However, neurons designs that interface with RRAMs need sufficient drive capability and additional circuitry for enabling in-situ learning[36,119,147].

Opamp-based neuron designs were introduced in[118,141] and designed to drive a large RRAM fan-out. Event-driven LIF neurons were demonstrated in 180nm CMOS which drove 1R RRAM crosspoint arrays with in situ STDP-based learning as illustrated in Fig. 10[23,41,148,149]. Here, the CBRAMs were integrated using wire bonding with the CMOS neurons and a neural circuit with associative learning (i.e. a Pavlov's Dog experiment) was demonstrated. In order to accommodate a wide variety of material stack used in the CBRAM, the neuron spike voltage and temporal profile parameters were designed to be digitally programmable. This neuron design had a bias current of $13\mu A$ in the integration mode and dynamically-biased with $56\mu A$ in the firing mode. The neuron's class-AB output stage could source up to 1.4mA current to drive an equivalent RRAM load of 10Ω with a power efficiency of 97%[41,148,149].

1T1R synapses not only significantly relax current sourcing requirement for the neurons, but also enable digital CMOS drive (i.e. with full-swing output between 0 and V_{DD}) as seen in Fig. 7. Fig. 11 shows an event-driven neuron that is adapted for the 1T1R crosspoint arrays[124]. The neuron operates asynchronously in two event-driven modes with a shared opamp– *integration* and *firing*. The neuron nominally operates in the integration mode, where the opamp is biased with very low bias current and configured as a leaky integrator with the virtual ground at the rest potential $V_{rst} = V_{DD}/2$, V_{DD} being the supply voltage. The incoming weighted current spikes, $i_i = \sum_j w_{ij} \cdot v_{pre,j}(t)$, are integrated on the capacitor C_m resulting in the membrane voltage

$$V_{m,j}(t) = V_{rst} + \sum_i \frac{G_{m,ij}}{C_{m,j}} \int_0^t v_{pre,i}(t) \otimes h(t) \cdot dt \tag{10}$$

Here, $h(t) = e^{-\frac{t}{\tau_{lk}}} u(t)$ is the impulse response of the LIF circuit during integration mode where $\tau_{lk} = R_{lk} C_{m,j}$ is the leak time-constant; R_{lk} is realized using the MOSFET, M_{lk}, biased in triode.

An asynchronous comparator is used to compare the membrane potential, $V_{m,j}$, with the threshold voltage, V_{thr}. When a positive crossing occurs, the neuron switched into the fire mode where its reconfigured as a voltage follower/buffer. Concurrently, an output spike with full digital levels is created and propagated forward.

TABLE II. Performance comparison of recent CMOS integrate and fire neurons.

Design	Architecture	Technology	Synapse Type	Spike Rate	In-situ Learning	Power	Energy-efficiency (/spike/synapse)	Area (μm^2)
indiveri et. al. 2006[?]	Subthreshold	$0.35\mu m$	CMOS, Bistable	200Hz	✓	-	90pJ	2,573
Basu and Hasler 2010[?]	Subthreshold	$0.35\mu m$	Floating Gate	100Hz	✗	1.74nW	17.4pJ	2,740
Cruz-Albrecht et. al. 2012[140]	Capacitive	90nm	CMOS, Dynamic	100Hz	✓	40pW	400fJ	442
Wu et. al. 2015[141]	Opamp	$0.18\mu m$	1R array†	0.1-1MHz	✓	$23.4\mu W$ $95.4\mu W$‡	9.3pJ	12,100
Sahoo 2017[142]	Ring VCO	65nm	None	0.4-1.5MHz	✗	-	-	-
Larras et. al. 2017[143]	Current summing	65nm	Digital, Binary	-	✗		7fJ	41,820
Sourikopoulos et. al. 2017[144]	Subthreshold	65nm	None	25kHz	✗	100pW	4fJ	35
Saxena 2020[124]	Opamp	$0.18\mu m$	1T1R array	0.1-1MHz	✓	$16.2\mu W$ $64.8\mu W$‡	8.1fJ△ 40fJ	5,625

† An LRS resistance of $R_{LRS} = 1k\Omega$ is assumed.
‡ Opamp is dynamically biased with higher current during the firing mode.
△ Energy-efficiency when used in inference mode only.

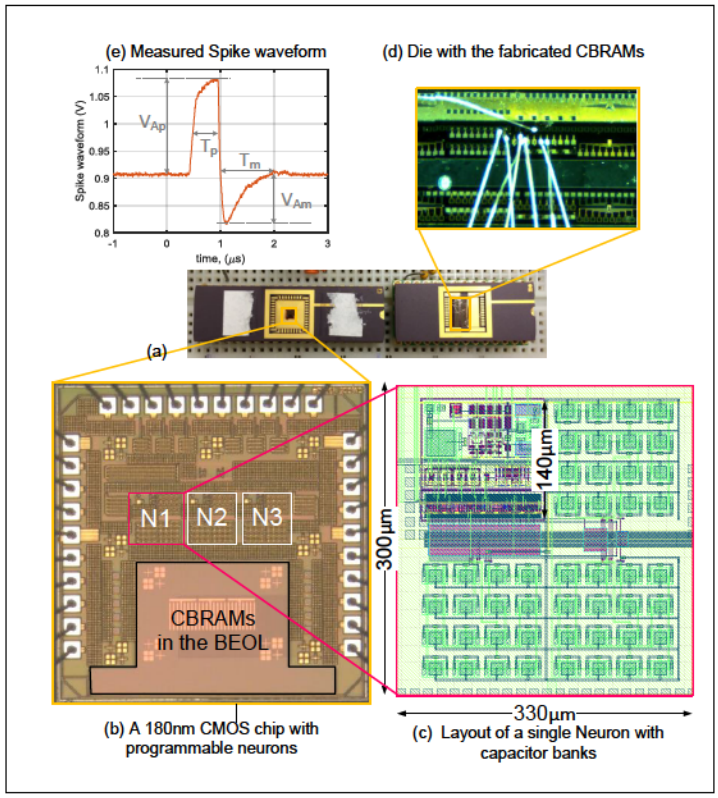

FIG. 10. (a) An opamp-based event-driven integrate and fire neuron to drive 1R CBRAM synapses as described in Refs.[41,141,149], (b) a 180nm CMOS chip with LIF neurons, (c) Layout of a single neuron with capacitor banks, (d) CBRAM devices as described in Ref.[93], (e) An output spike generated by the neuron.

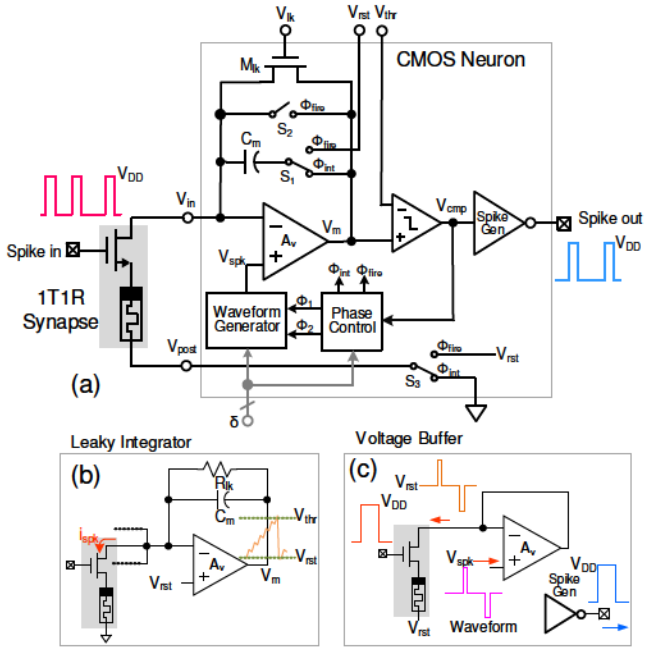

FIG. 11. (a) Schematic of the event-driven CMOS neuron for 1T1R array architecture as described in Ref.[124], (b) Neuron in the integration mode, (c) Neuron in the firing mode where a weight update can take place.

If on-chip learning is enabled then the waveform seen in Fig. 9 is generated by a switched-capacitor circuit. The phase control circuit generates the strobes ϕ_{int} and ϕ_{fire} for the switched-capacitor circuits, and ϕ_1 and ϕ_2 for waveform generation. The waveform parameters can be digitally configured for a specific RRAM material stack and can be changed on the fly. The waveform drives the positive opamp input while the opamp is configured as a voltage follower. The voltage follower provides the sufficient drive current for the RRAMs that are in the STDP window where pre- and post-spikes overlap[124]. After the fire phase is concluded, the neuron enters a refractory period, where it is reconfigured back in the integration mode and $V_{m,j}$ is reset to V_{rst}.

The CMOS neuron from Fig. 11 was designed 180nm CMOS process with a $V_{DD} = 1.8V$[124]. A compact model for the RRAM[150] was implemented in Verilog-A. The total bias current was $9\mu A$ and and energy-efficiency of 40fJ/spike/synapse. Compared to prior neurons that were designed to drive 1R synapses[141], this 1T1R design only drives a the small input capacitance of the access transistor during inference, and resistive load only dur-

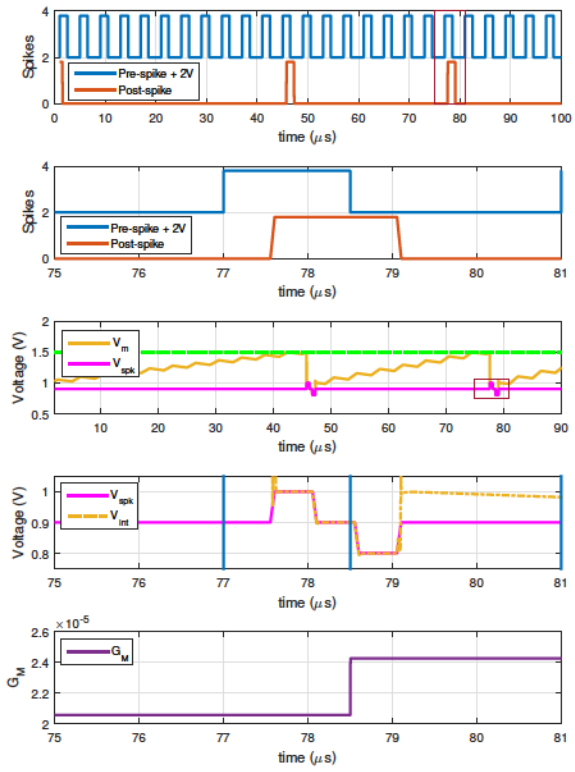

FIG. 12. Plots generated from a transient simulation of the 180nm CMOS neuron for driving 1T1R synapses seen in Figure 11. The asynchronous input spikes are integrated into the membrane potential, V_m. When V_m crosses V_{thr}, the neuron fires and a full-swing output spike is generated. Also, a spike waveform, V_{spk}, is generated and applied across the 1T1R synapse, and thus the RRAM device, by the opamp which is configured as a voltage follower. In this case, the conductance of the synapse is increased as the positive portion of the spike waveform overlaps with the pre-spike pulse.

ing the training/learning phases.

A performance comparison of recent integrate and fire neurons from the literature is provided in Table II. As evidenced by this comparison, very few architectures have addressed the circuit interfacing challenges with the RRAM devices. There is significant room for reduction in neuron energy consumption by employing scaled CMOS nodes, and techniques including adaptive biasing (with static bias currents in the nAs), relaxing the transition times on the program/erase pulses and power gating. However, the designs will continue to co-evolve with spike-based learning algorithms.

V. NEUROMORPHIC DEEP LEARNING ALGORITHMS

Realization of DNNs in a low-power hardware has been of great interest in applications where neural network processing needs to occur close to the sensors with real-time inference and with minimal reliance on the Cloud infrastructure. This is driven by two divergent paradigms. The first view is driven by the need to demonstrate competitive performance on real-world applications compared to Backprop-based standard DNNs. The second view is of exploring the underlying mechanism behind distributed and continual learning in biological brains, which can learn from small amount of data in an unsupervised or semi-supervised manner. A brief overview of evolving neuromorphic learning algorithms is presented in this section and their performance is compared in Table III.

A. Transfer Learning

A straightforward approach is to first train a DNN comprised of differentiable non-spiking neurons using Backprop, regularization methods such as Dropout, and optimization techniques including batch normalization[3]. Then the DNN is converted to its equivalent SNN by employing rate based coding, scaling the weights based on the spiking activity, and adjusting the thresholds to minimize absence of spikes or neuron saturation due to over-spiking[156,171]. There is a perceptible performance drop in classification accuracy in the DNN to SNN conversion[156–158,171]. Transfer learning has also been used to initialize a deep SNN which is then further trained using Backprop[172]. To facilitate transfer learning, a neuromorphic IC should have a mechanism to download a pre-trained model and program the on-chip synapses to the desired weights. Furthermore, these weights decay over time due to resistance drift and a *drift compensation* scheme is needed to maintain model accuracy over longer time scales[173]. Transfer learning has achieved highest accuracy of 99.44%[158] for the MNIST handwritten digits dataset compared to the best in-class DNN accuracy of 99.79%[151]. For CIFAR-10 dataset, this gap is within 2% of the Alexnet DNN.

B. Semi-supervised Learning

Spike-based unsupervised or semi-supervised learning algorithms are based on the neural-inspired WTA motifs of LIF neurons with inhibition and competition. Their ability to learn spatiotemporal patterns using STDP was first studied using software simulations[174] and was then applied to vision tasks[31,175]. Subsequently, analytical modeling showed that WTA neurons with STDP learning realize a spiking version of the Expectation Maximization algorithm[34]. This two layer motif learns from a small

TABLE III. Comparison of deep SNN Algorithms and their benchmarking against the highest performing non-spiking DNNs.

Reference	Algorithm/Neuron Type	Network	Dataset	Accuracy
Standard Deep Neural Networks				
Simard et. al. 2003[151]	Backprop with ReLU	784-20c5-2s-100-10o	MNIST	99.6%
Wan et. al. 2013[152]	Backprop with ReLU and DropConnect	-	MNIST	99.79%
Krizhevsky et. al. 2012[153]	Alexnet: Backprop with ReLU	5-layer ConvNet	CIFAR-10	88.91%
He et. al. 2016[154]	ResNet: Backprop with ReLU	ResNet-1001	CIFAR-10	95.08%
VGGNet 2017[155]	Backprop with ReLU	16-layer ConvNet	ImageNet$^{\triangledown}$	93.2%
SNNs with Transfer Learning				
Diehl et. al. 2015[156]	Fully connected with Rate-based IFN	784-1200-1200-10o	MNIST	98.68%
	ConvNet with Rate-based IFN	784-12c5-2s-64c5-2s-10o		99.14%
Hunsberger et. al. 2016[157]	ConvNet with Rate-based LIFN	784-16c5-2s-64c5-2s-10o	MNIST	99.1%
		Alexnet	CIFAR-10	82.95%
		-	ImageNet	76.2%
Rueckauer 2017[158]	ConvNet with Rate-based LIFN	LeNet	MNIST	99.44%
		Alexnet	CIFAR-10	88.82%
		VGG-16	ImageNet$^{\triangledown}$	84.86%
		Inception-V3	ImageNet$^{\triangledown}$	92.04%
Sengupta et. al. 2019[159]	ResNet with Rate-based LIFN	ResNet-20	CIFAR-10	87.46%
		ResNet-34	ImageNet	86.43%
Unsupervised Learning SNNs				
Diehl & Cook 2015[160]	STDP-WTA, LIFN with inhibition	784-1600-10o	MNIST	95%
Kheradpisheh et. al. 2016[161]	STDP-WTA, IFN with latency coding	784-30c5-2s-100c5-2s-10o	MNIST	98.4%
Vaila et. al. 2019[162]	Binarized STDP-WTA, Surrogate gradients	784c5-2s-30c5-s7-500-70-10o	MNIST	98.49%
			EMNIST	85.3%
Spike-based Backpropagation				
Neftci et. al. 2016[163]	Event-driven RBP with Rate-based IFN	784-500-10o	MNIST	97.04%
Lee et. al. 2016[164]	Backprop with Rate-based IFN	784-800-10o	MNIST	99.31%
O'Connor et. al. 2017[165]	Backprop with Delta coding	784-200-200-10o	MNIST	98.36%
Kulkarni et. al. 2018[166]	NormAD with IFN	784-12c3-10o	MNIST	98.17%
Mostafa 2018[167]	Backprop with Temporal coding	784-400-400-10o	MNIST	97.55%
Shrestha et. al. 2018[168]	SLAYER with Rate-based IFN	784-12c5-2s-64c5-2s-10o	MNIST	99.36%
Mostafa et. al. 2018[169]	DNN with Synthetic gradients, ReLU†	784-1000-1000-1000-10o	MNIST	98.7%
		ConvNet3$^{\triangle}$	CIFAR-10	89.1%

† This work uses non-spiking ReLUs but can be adapted to SNNs using rate-based or temporal-coding[170].
$^{\triangle}$ Network for CIFAR-10 is 32x32x3-96c5-s3-128-c5-s3-256c5-s3-2048-2048-10o.
$^{\triangledown}$ Top-5 accuracy on ImageNet dataset.

number of samples and the SNN weights converge to the log probability of the patterns in the training dataset[34].

The above mentioned two-layer SNN, without a hidden layer, was applied to the MNIST handwritten digits recognition task[41,160]. Using competitive learning, each neuron learned to fire on distinct inputs. The output labels are either assigned after the learning is completed[160] or only the intended output neurons are allowed to fire[41]. This semi-supervised SNN achieved a classification accuracy of 94% for four digits and 83% on all the ten digits (with 10 output neurons) with around 1000 training samples for each image label[41]. Higher classification accuracy was demonstrated by using a large number of competing neurons ($\approx$ 5000) leading to a maximum classification accuracy of 95%[160].

The WTA motifs can be organized in a ConvNet architecture as shown in Fig. 13 where only one neuron per kernel is allowed to fire across all the feature maps[161,177]. Stacking of these spiking ConvNet motifs to improve classification performance was explored next[161,162,177,178]. While the ConvNet layers learn in an unsupervised manner, a fully-connected read-out layer which is trained using Backprop is employed as seen in Fig. 13. It was found that greedy stacking of more than two unsupervised learning ConvNet layers didn't improve the classification accuracy. In essence, each WTA with STDP layer can be thought of as if its performing unsupervised clustering over the input feature space. This learning occurs with very few samples. However, in the absence of a mechanism to assign credit across layers based on the output classification error, the resulting accuracy doesn't improve by increasing the SNN depth. These deep semi-supervised SNNs have demonstrated a maximum accuracy of 98.5% for the MNIST handwritten digit dataset[161,162,177–179], and have been shown to be suitable for incremental learning of tasks[180].

C. Backpropagation-based Learning

In spite of the desirable features of the semi-supervised learning SNNs, there is a classification accuracy gap between semi-supervised SNNs and Backprop-trained DNNs. As a result, there has been sustained interest in adapting Backprop to SNNs[163]. Backprop, along with the ConvNet layers, is the workhorse for deep learning and minimizes the output classification error by propagating error gradients backward from the output layer, L, to the lower network layers, $1 \leq l < L$. This is analytically described by the four Backprop equations below[3]

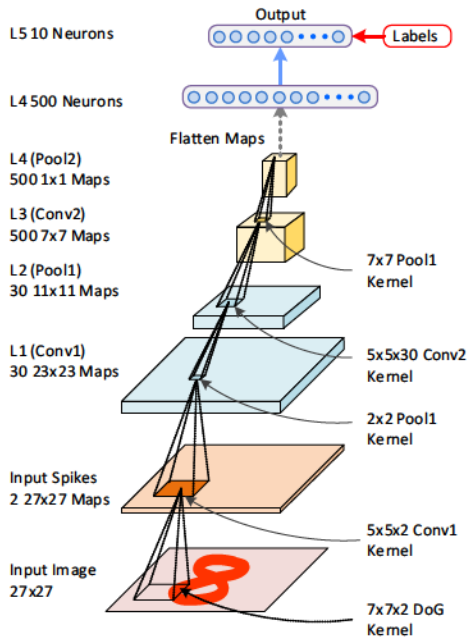

FIG. 13. A Spike-based ConvNet for MNIST handwritten digits dataset: 27×27×2-30c5-2s-500c7-7s-10o as described in Ref.[176]. Edge detection is performed using On-center and Off-center Difference of Gaussian (DoG) kernels to result in a 27x27x2 image.

$$\delta^L = \nabla_a \mathcal{C} \odot \sigma'(z^L) \tag{11a}$$

$$\delta^l = ((W^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \tag{11b}$$

$$\frac{\partial \mathcal{C}}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \tag{11c}$$

$$\frac{\partial \mathcal{C}}{\partial \theta^l_j} = \delta^l_j \tag{11d}$$

Here, $\mathcal{C}$ is a formulation of the output loss, or cost, function and given by[3]

$$\mathcal{C} = \frac{1}{2N} \sum_x \|y(x) - a^L(x)\|^2 \tag{12}$$

where N is the batch size, and x and y represent a sample and the corresponding label in the dataset respectively. Furthermore, a^l_j represents the activation for the i^{th} neuron in the l^{th} layer ($1 \leq l \leq L$), $\sigma(\cdot)$ is the equivalent activation function, δ^l_j is the backpropagated error for neuron j in layer l, and $\theta^l_j \equiv V_{rst}$ is the spike threshold for the neuron. Also, $z^l_j = \sum_j w^l_{jk} \nu^{l-1}_k$ is analogous to the neuron membrane potential, $V_{m,j}$. SNNs can either be formulated using rate-based or spike latency coding[167,181].

For every training batch, each of the network weights are updated using the gradients as[163]

$$w^{(t+1)}_k = w^{(t)}_k - \eta \frac{\partial \mathcal{C}}{\partial w_k} \tag{13a}$$

$$\theta^{(t+1)}_j = \theta^{(t)}_j - \eta \frac{\partial \mathcal{C}}{\partial \theta_j} \tag{13b}$$

where η is the learning rate. Since the activation of spiking neurons is discontinuous, direct computation of gradients is not feasible. As a result, equivalent linearized or stochastic differentiable neuron models have been derived to obtain a continuous activation function, $\sigma()$, with a well defined derivative, $\sigma'()$[163,168].

1. Challenges with Spike-based Backprop

Since Backprop has evolved using von Neumann architectures, it assumes that the network-wide weights, neuron activations, and their derivatives are always accessible from a high-density memory. This memory has a latency and energy cost associated with data access which percolates through the von Neumann bottlenecks[163,170]. Neuromorphic computers, on the other hand, aim to minimize this back and forth shuttling of data by performing localized computing inside the memory itself; however, the nature of data flow associated with Backprop presents the following architectural challenges[23,163]:

1. **Weight Transport Problem**: In order to compute the weight updates at layer l using Eq. 11b, the transpose of the weight matrix for layer $(l+1)$, $(W^{l+1})^T$, must be available while evaluating weights connecting layers $(l-1)$ and l, which poses challenges for hardware design.
2. **Non-concurrence**: Data flow must alternate between forward and backward passes during each minibatch which limits learning on real-time streaming data.
3. **Differentiability and Precision:** Derivatives need to be computed with high-precision or approximated to simpler functions.
4. **Temporal Credit Assignment Problem**: During training of a DNN as shown in Fig. 13, the network layers undergo a forward pass and then wait for the gradients to be propagated in the reverse direction. This poses a temporal credit assignment problem where future errors are needed to update weights based on current spike correlations. Consequently, the lower layers in the network are frozen till the weight update in the backward pass takes place[182].

2. Recent Advances in Spike-based Backprop

Several techniques have appeared in spike-based Backprop adaptations where the constraints of standard Back-

prop are relaxed to simplify neuromorphic hardware design. Random Backprop (RBP) or feedback alignment (FA) algorithm eliminates the symmetry constraint[183]. Here, instead of $(W^{l+1})^T$ in the weight update computation in Eq. 11c, a fixed random matrix, B^{l+1} is used. This replaces Eq. 11c by[183]

$$\delta^l_{FA} = (B^{l+1}\delta^{l+1}) \odot \sigma'(z^l) \tag{14}$$

Switching to the fixed matrix simplifies the symmetry constraint in the architecture without incurring significant decrease in classification accuracy[183]. Random Backprop was adapted to event-driven neurons in an early demonstration of deep learning in SNNs[163].

In the Backprop Eqs. 11a to 11d seen earlier, a continuous, and thus differentiable, model for the neuron activation function, $\sigma(\cdot)$, was assumed. Recent results in deep SNNs have developed continuous models for the LIF neurons by considering signal noise to soften the hard decision thresholds. In one of these differential neuron models, the derivative of spiking activation was approximated as:

$$\sigma'(z^l_j) = \alpha e^{-\beta|z^l_j - \theta|} \tag{15}$$

where, α and β are model constants, and θ is the neuron threshold[168].

By combining RBP with event-driven spikes and approximate rate-based gradients[163], an accuracy of 97.4% for the MNIST dataset was obtained with a three-layer fully-connected network. In another rate-coding SNN, the WTA structure was preserved in the weight update rules and resulted in an MNIST accuracy of 98.71%[164]. In latency or time-coded SNNs, SpikeProp[181] was an early work where the post neuron spike times were linearized to compute gradients. Another learning rule called *normalized approximate gradient descent (NormAD)* was proposed for temporally-coded SNNs, where an approximate gradient was defined by linearizing the membrane potential as a function of input spikes[166,184]. In a more recent work, the explicit spike delays were employed to compute gradients without any linearization[167]. This work resulted in a reduced MNIST classification accuracy at 97.55%.

A recent SNN training algorithm, called SLAYER, adapted spike-based Backprop where learning occurs in both the weights as well as the axonal delays using the stochastic exponential neuron model seen in Eq. 15[168]. In this work, a temporal credit assignment scheme is employed where the error is backpropagated through time. The algorithm was implemented in a GPU and achieved the highest SNN classification accuracy for MNIST dataset at 99.44%[168]. The requirement of propagating error back in time was solved by the SuperSpike[185] algorithm. SuperSpike employs a three-factor learning rule with synaptic eligibility traces to solve the temporal credit assignment problem[170].

From the comparison in Table III, it can be seen that transfer learning achieves the highest accuracy for SNNs which comes very close to the classification performance for the AlexNet-size DNNs. Steady progress has been made in adapting Backprop to SNNs with the recently reported algorithms demonstrating competitive performance on the MNIST and CIFAR-10 datasets while relaxing the hardware bottlenecks associated with the standard Backprop.

VI. NVM-BASED NEUROMORPHIC ICS

Early analog neuromorphic ICs comprised of low-power subthreshold analog neuron and synapse circuits and were primarily intended for emulating ion channel kinetics in biological neural circuits[21,46]. Mixed-signal ICs employed SRAM with DACs to realize synapses along with analog LIF neurons[49,191]. These were scaled to wafer-level SNN implementations and demonstrated several neurobiological as well as neuromorphic computing tasks[192,193].

Progress in digital neuromorphic hardware platforms has led to the realization asynchronous event-driven (as opposed to clock driven) computing ICs that communicate information on and across the chips using digital spikes. The most pertinent examples of digital neuromorphic chips are IBM's TrueNorth chip[44], the recent Loihi chip from Intel[45], and the two versions of SpiNNaker systems from the European Brain Project[194–196].

Table IV presents NVM-based Neuromorphic ICs in the literature along with their performance benchmarking. Development of in-memory computing neuromorphic ICs initially focused on small-scale NVM device arrays for characterization of device switching, multi-level or analog states, variability, retention and endurance[41,117,137,197–199]. In several of these works, device results were extrapolated to DNNs or SNNs which were entirely simulated in software.

A. NOR Flash Architectures

Flash-based neuromorphic architectures have been studied for the past several years[18]. However, the recent interest in NVM-based VMMs led to their first hardware demonstration using established NOR Flash arrays integrated in a standard CMOS technology, along with the array programming and read circuitry[187]. In this work, a three-layer analog neural network was demonstrated using 180nm NOR Flash array[187]. The chip was programmed using a transfer learning approach with 6-bit analog precision and demonstrated an MNIST accuracy of 94.7%. An important observation was that the cell conductances decayed by around 13% over the 7 months storage period, however, the classification accuracy had minimal degradation and it remained above 94%.

TABLE IV. Comparison of NVM-based Neuromorphic ICs for Deep Learning.

Design	Technology	Architecture	Synapse Resolution	On-chip learning	Network Size	Dataset, Classf. Acc.	Energy-efficiency
Yu 2016[186]	130nm CMOS + TaO_x/HfO_2 OxRAM	Binary VMM	1-bit	✗	400-200-10o	MNIST, 96.5%	-
Guo 2017[187]	180nm NOR Flash	Analog VMM	6-bit Analog	✗	784-256c5-10o	MNIST, 94.7% CIFAR-10, 84.8%†	20nJ/inference
Li 2018[188]	2μm CMOS + Ta/HfO_x OxRAM	R array	Analog	✓	64-10o	8x8 UCI, 91.7%	-
Wang 2018[189]	Ta/HfO_x OxRAM	1T1R array	Analog	✓	8-8o	Custom patterns	-
Cai 2019[69] & Correll 2019[139]	180nm CMOS + WOx RRAM	1R array, Mixed-signal	Analog	✓	54x108 array	5x5 images	8.5GOPS/W
CEA-Leti 2019[115]	130nm CMOS + Ti/HfO_x OxRAM	1T1R, SNN	1-bit	✗	784-10o	MNIST, 84%	180pJ/SynOp
Xue 2019[190]	55nm CMOS + RRAM	VMM, 1T1R	3-bit Digital	✗	1Mb Macro	CIFAR-10, 85.52%‡	53.17TOPS/W
Hirtzlin 2020[131]	130nm CMOS + Ti/HfO_x OxRAM	2T2R	1-bit	✗	Off-chip	MNIST, 98.3%† CIFAR-10, 87.5%† ImageNet, 69.7%†	20-30pJ/SynOp
Nandakumar 2020[173]	90nm CMOS + GST PCRAM	2T1R, LIFN	7-bit Analog	✓	784-10o	MNIST, 70%	-
Liu 2020[133]	130nm CMOS + RRAM	VMM, 2T2R	1 to 8-bit Digital	✗	784-100-10o	MNIST, 94.4%	78.4TOPS/W
Wan 2020[146]	130nm CMOS + TaO_x RRAM	IFN, 1T1R	1-bit	✗	225-60	MNIST RBM	74TOPS/W
Xue 2020[114]	22nm CMOS + RRAM	VMM, 1T1R	4-bit Digital	✗	1Mb Macro	CIFAR-10, 90.19%‡ CIFAR-100, 64.15%‡	121.38TOPS/W

† Simulated result.
‡ Result is using off-chip computation.

B. CMOS-RRAM Architectures

Initial demonstrations focused on very simple pattern learning tasks using small-scale RRAM memory arrays. For example, synapses were interfaced with discrete electronic circuits to demonstrate a small-scale network[200]. In an early work on event-driven RRAM-compatible neuron design, a 3-neuron associative SNN was experimentally demonstrated[148], and then adapted to 8×8 UCI handwritten digit dataset[41].

These were followed by in-silicon demonstration of shallow two-layer neural networks. For example, online learning of binarized neural networks was demonstrated using TaO_x/HfO_2 RRAMs in 130nm CMOS and resulted in an MNIST accuracy of 96.5%[186]. In another work, in-situ learning in a two-layer network was demonstrated using Ta/HfO_x memristor array integrated with 2μm CMOS[188]. However, in this work, the neuron activations were simulated in software[188]. Furthermore, 8×8 Ta/HfO_x/Pd 1T1R arrays were demonstrated to learn basic patterns under unsupervised training[189].

With the recent integration of RRAMs with CMOS transistors in a foundry process, neuromorphic ICs with medium to large-scale integration of circuits with RRAMs have begun to appear. A majority of these works target inference-only applications by leveraging the high density of RRAM arrays with analog-domain multiply and accumulate (MAC) operations[114,115,131,133,146,186,190]. Among the inference-only demonstration chips, either binarized weights were employed, or parallel RRAM cells were used to emulate a multibit synapse. In the latter case, unit-weighted binary 1T1R cells were used as a current-DAC to realize multi-bit synapses[114,131,133,146,190].

As an example of large-scale integration of OxRAMs with 130nm CMOS, a 2Kb differential binary RRAM (2T2R) array with integrated column sense-amps was fabricated, and on-chip inference was exhibited for MNIST, CIFAR and ImageNet datasets with competitive classification performance[131]. In another recent work[133] using a similar technology, binarized 2T2R synapses were used for signed weights in order to demonstrate a multi-bit VMM. Again, in this work multi-bit weights were realized by combining several 2T2R cells in parallel. The weighted currents were integrated and digitized using a successive approximation register (SAR) ADC. This chip-scale demonstration fully-integrated a 784-100-10 network with an FPGA-based back-end, and demonstrated 94.4% MNIST accuracy[133]. Overall, a full-chip integration that demonstrates true multilevel-cell 1T1R synapses for on-chip inference is yet to be seen (other than the attempts in the works[69,139]).

So far, only a few designs have attempted fully-integrated on-chip learning where the challenges associated with analog synapses need to be addressed[69,139]. This CMOS-RRAM IC prototype incorporates 54×108 WOx RRAM crossbar array integrated with 180nm CMOS[69,139]. The chip also includes arrays of 6-bit DAC and 13-bit column ADCs and a RISC processor for digital backend. The row DAC produces voltage pulses of fixed width proportional to the input value. These DAC pulses are weighted by the RRAM-based VMM and then integrated in charge-domain on the integrating-type column ADCs, thus realizing a mixed-signal VMM with digital input and output vectors.

VII. DISCUSSION AND CONCLUSION

From the discussion in Section VI, it is evident that fully CMOS-based digital and mixed-signal neuromorphic ICs feature very high-level of system integration and functionality. This is due to the maturity of design, modeling and verification infrastructure for CMOS technology. However, CMOS realizations either exhibit lower neurosynaptic density, employ binarized volatile synapses, and can be limited by a von Neumann bottleneck.

While digital neuromorphic ICs have made significant progress in low-power realizations of deep neural networks, NVMs have a role to play in another order of magnitude improvement in neurosynaptic density and energy-efficiency. Among the NVMs, NOR Flash-based architectures have demonstrated multi-layer neural networks for inference, all due to the maturity of floating-gate devices, proven multilevel cell capability, and their longer retention times.

As seen in Table IV, RRAM and PCRAM-based neuromorphic inference ICs have shown steady progress, while on-chip training architectures are still in their infancy. Hybrid CMOS-RRAM inference SoCs using binarized RRAMs have demonstrated higher neurosynaptic density, competitive on-chip classification accuracies, and higher energy-efficiency approaching 100 TOPS/W. RRAMs also promise in-situ training capability due to their significantly higher reported endurance approaching 10^9 write cycles. However, several device-level challenges such as the controllability of multilevel states and state drift need to be addressed. Resistance state drift in RRAM synapses degrades the classification performance of the neural network models, that rely on multi-bit synapses, only in a few hours and thus restoration of the states will have to be addressed at the circuit as well as algorithmic level.

As far as algorithms are concerned, a direct adaptation of Backprop to SNNs may not be the actual algorithm responsible for cognitive 'computation' occurring in the biological brains. Nevertheless, it provides an intermittent solution to embedded AI applications desired by the computing community. Needless to say, development of learning algorithms for SNN is a promising area of research and together with developments in the field of computational neuroscience, it may lead to better understanding of brain computation. However, going forward with the development of large-scale neuromorphic computing architectures, these algorithms will synergistically evolve by accommodating the realistic behavior of synaptic devices and by alleviating the hardware bottlenecks that arise when deep learning algorithms are mapped to in-memory computing hardware.

ACKNOWLEDGMENTS

The author would like to thank Dr. Tomas Wu for CBRAM characterization, Prof. Maria Mitkova for providing CBRAM samples fabricated by Dr. Muhammad Rizwan Latif, Prof. John Chiasson and Ruthvik Vaila for the results on unsupervised learning in SNNs.

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

[1]Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).
[2]I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT press, 2016).
[3]M. Nielsen, *Neural Networks and Deep Learning*, 1st ed. (2017).
[4]M. H. Ionica and D. Gregg, IEEE Micro **35**, 6 (2015).
[5]P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith, arXiv preprint arXiv:1812.01739 (2018).
[6]M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, in *OSDI*, Vol. 16 (2016) pp. 265–283.
[7]A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, (2017).
[8]A. Fuchs and D. Wentzlaff, in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)* (IEEE, 2019) pp. 1–14.
[9]IRDS, "International Roadmap for Devices and Systems (IRDS): More Moore Technologies," (2017).
[10]R. Waser, R. Dittmann, G. Staikov, and K. Szot, Advanced materials **21**, 2632 (2009).
[11]D. Ielmini and H.-S. P. Wong, Nature Electronics **1**, 333 (2018).
[12]A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, Nature Nanotechnology , 1 (2020).
[13]Z. Wang, H. Wu, G. W. Burr, C. S. Hwang, K. L. Wang, Q. Xia, and J. J. Yang, Nature Reviews Materials , 1 (2020).
[14]W. Zhang, R. Mazzarello, M. Wuttig, and E. Ma, Nature Reviews Materials **4**, 150 (2019).
[15]A. Sebastian, M. Le Gallo, and E. Eleftheriou, Journal of Physics D: Applied Physics **52**, 443002 (2019).
[16]Q. Xia and J. J. Yang, Nature materials **18**, 309 (2019).
[17]M. Pfeiffer and T. Pfeil, Frontiers in neuroscience **12**, 774 (2018).
[18]J. Hasler and H. B. Marr, Frontiers in neuroscience **7**, 118 (2013).
[19]C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, arXiv preprint arXiv:1705.06963 (2017).
[20]Z. You and S. Wei, "White Paper on AI Chip Technologies," (2018).
[21]S.-C. Liu, *Event-Based Neuromorphic Systems* (John Wiley & Sons, 2015).
[22]S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, IEEE transactions on biomedical circuits and systems **12**, 106 (2017).
[23]V. Saxena, X. Wu, I. Srivastava, and K. Zhu, Journal of Low Power Electronics and Applications **8**, 34 (2018).
[24]W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, Nature **383**, 76 (1996).
[25]G.-q. Bi and M.-m. Poo, The Journal of neuroscience **18**, 10464 (1998).
[26]G.-q. Bi and M.-m. Poo, Annual review of neuroscience **24**, 139 (2001).
[27]P. J. Sjöström, G. G. Turrigiano, and S. B. Nelson, Neuron **32**, 1149 (2001).

[28]N. L. Golding, N. P. Staff, and N. Spruston, Nature **418**, 326 (2002).
[29]Y. Dan and M.-m. Poo, Neuron **44**, 23 (2004).
[30]Y. Dan and M.-M. Poo, Physiological reviews **86**, 1033 (2006).
[31]T. Masquelier and S. J. Thorpe, PLoS computational biology **3**, e31 (2007).
[32]U. Weidenbacher and H. Neumann, in *Perception in Multimodal Dialogue Systems* (Springer, 2008) pp. 123–131.
[33]H. Lee, *Unsupervised feature learning via sparse hierarchical representations* (Stanford University, 2010).
[34]B. Nessler, M. Pfeiffer, L. Buesing, and W. Maass, PLoS computational biology **9**, e1003037 (2013).
[35]D. Sterratt, B. Graham, A. Gillies, and D. Willshaw, *Principles of computational modelling in neuroscience* (Cambridge University Press, 2011).
[36]R. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, in *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on* (IEEE, 2014) pp. 1564–1567.
[37]A. van Schaik, Neural Networks **14**, 617 (2001).
[38]E. M. Izhikevich, IEEE Transactions on neural networks **14**, 1569 (2003).
[39]A. van Schaik, C. Jin, A. McEwan, and T. J. Hamilton, in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (IEEE, 2010) pp. 4253–4256.
[40]G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, *et al.*, Frontiers in neuroscience **5**, 73 (2011).
[41]X. Wu, V. Saxena, and K. Zhu, IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS) **5**, 254 (2015).
[42]Y. Bengio, T. Mesnard, A. Fischer, S. Zhang, and Y. Wu, arXiv preprint arXiv:1509.05936 (2015).
[43]J. Sjöström and W. Gerstner, Scholarpedia **5**, 1362 (2010).
[44]P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Rish, R. Manohar, and D. S. Modha, Science Magazine **345**, 668 (2014).
[45]M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, and H. Wang, IEEE Micro **38**, 82 (2018).
[46]K. Boahen, in *International Conference of the IEEE Engineering in Medicine and Biology Society* (2006).
[47]G. Indiveri, E. Chicca, and R. Douglas, IEEE Transactions On Neural Networks **17**, 211 (2006).
[48]V. Saxena, X. Wu, and K. Zhu, in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (2018) pp. 1–5.
[49]T. Pfeil, T. C. Potjans, S. Schrader, W. Potjans, J. Schemmel, M. Diesmann, and K. Meier, arXiv preprint arXiv:1201.6255 (2012).
[50]E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs, Frontiers in neuroscience **7** (2013).
[51]S. Brink, S. Nease, and P. Hasler, Neural Networks (2013).
[52]A. Chen, Solid-State Electronics **125**, 25 (2016).
[53]D. Jana, S. Roy, R. Panja, M. Dutta, S. Z. Rahaman, R. Mahapatra, and S. Maikap, Nanoscale Research Letters **10**, 188 (2015).
[54]S. Dutta, C. Schafer, J. Gomez, K. Ni, S. Joshi, and S. Datta, Frontiers in Neuroscience **14** (2020).
[55]D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, Nature **453**, 80 (2008).
[56]A. Rothenbuhler, T. Tran, E. Smith, V. Saxena, and K. Campbell, Journal of Low Power Electronics and Applications **3**, 174 (2013).
[57]D. Kuzum, R. G. Jeyasingh, B. Lee, and H.-S. P. Wong, Nano letters **12**, 2179 (2011).
[58]G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, K. Virwani, M. Ishii, P. Narayanan, A. Fumarola, L. L. Sanches, I. Boybat, M. Le Gallo, K. Moon, J. Woo, H. Hwang, and Y. Leblebici, Advances in Physics: X **2**, 89 (2017).
[59]Z. Sun, X. Bi, H. Li, W.-F. Wong, and X. Zhu, IEEE Transactions on Very Large Scale Integration (VLSI) Systems **22**, 1281 (2013).
[60]M. Jerry, P.-Y. Chen, J. Zhang, P. Sharma, K. Ni, S. Yu, and S. Datta, in *2017 IEEE International Electron Devices Meeting (IEDM)* (IEEE, 2017) pp. 6–2.
[61]J. J. Yang, D. B. Strukov, and D. R. Stewart, Nature nanotechnology **8**, 13 (2013).
[62]S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, and H.-S. P. Wong, Electron Devices, IEEE Transactions on **58**, 2729 (2011).
[63]D. Kuzum, S. Yu, and H. P. Wong, Nanotechnology **24**, 382001 (2013).
[64]S. Menzel, M. Waters, A. Marchewka, U. BÃƒÂ¶ttger, R. Dittmann, and R. Waser, Advanced Functional Materials **21**, 4487 (2011).
[65]R. Waser, R. Dittmann, G. Staikov, and K. Szot, Advanced Materials **21**, 2632 (2009).
[66]M. Zhao, H. Wu, B. Gao, Q. Zhang, W. Wu, S. Wang, Y. Xi, D. Wu, N. Deng, S. Yu, H.-Y. Chen, and H. Qian, in *IEEE International Electron Devices Meeting (IEDM)* (IEEE, 2017) pp. 39–4.
[67]A. Benoist, S. Blonkowski, S. Jeannot, S. Denorme, J. Damiens, J. Berger, P. Candelier, E. Vianello, H. Grampeix, J. Nodin, E. Jalaguier, and L. Perniola, in *2014 IEEE International Reliability Physics Symposium* (IEEE, 2014) pp. 2E–6.
[68]X. Guo, F. M. Bayat, M. Prezioso, Y. Chen, B. Nguyen, N. Do, and D. B. Strukov, in *2017 IEEE Custom Integrated Circuits Conference (CICC)* (IEEE, 2017) pp. 1–4.
[69]F. Cai, J. M. Correll, S. H. Lee, Y. Lim, V. Bothra, Z. Zhang, M. P. Flynn, and W. D. Lu, Nature Electronics **2**, 290 (2019).
[70]D. Ielmini and R. Waser, *Resistive switching: from fundamentals of nanoionic redox processes to memristive device applications* (John Wiley & Sons, 2015).
[71]S. H. Jo, K.-H. Kim, and W. Lu, Nano letters **9**, 870 (2009).
[72]S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, Nano letters **10**, 1297 (2010).
[73]K. Seo, I. Kim, S. Jung, M. Jo, S. Park, J. Park, J. Shin, K. P. Biju, J. Kong, K. Lee, B. Lee, and H. Hwang, Nanotechnology **22**, 254023 (2011).
[74]Y. Li, Y. Zhong, L. Xu, J. Zhang, X. Xu, H. Sun, and X. Miao, Scientific Reports **3** (2013).
[75]R. Waser, R. Dittmann, G. Staikov, and K. Szot, Advanced materials **21**, 2632 (2009).
[76]R. Waser, D. Ielmini, H. Akinaga, H. Shima, H.-S. P. Wong, J. J. Yang, and S. Yu, Resistive Switching: From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications , 1 (2016).
[77]D. J. Wouters, in *Workshop on Memristive systems for Space Applications, Noordwijk, Netherlands* (2015).
[78]M. Kozicki, M. Balakrishnan, C. Gopalan, C. Ratnakumar, and M. Mitkova, in *Symposium Non-Volatile Memory Technology 2005.* (Institute of Electrical & Electronics Engineers (IEEE)).
[79]M. N. Kozicki and M. Mitkova, *Memory Devices Based on Mass Transport in Solid-state Electrolytes, Chapter 16 Nanotechnology. Volume 3*, edited by R. Waser (WILEY-VCHGmbH&Co KGaA and Weinheim, 2010).
[80]M. N. Kozicki, M. Mitkova, and I. Valov, "Electrochemical metallization memories," in *Resistive Switching* (Wiley-Blackwell, 2016) pp. 483–514.
[81]M. R. Latif, M. Mitkova, G. Tompa, and E. Coleman, in *2013 IEEE Workshop on Microelectronics and Electron Devices (WMED)* (IEEE, 2013) pp. 1–4.
[82]C. Nail, G. Molas, P. Blaise, G. Piccolboni, B. Sklenard, C. Cagli, M. Bernard, A. Roule, M. Azzaz, E. Vianello, *et al.*, in *2016 IEEE International Electron Devices Meeting (IEDM)* (IEEE, 2016) pp. 4–5.

[83] M. R. Latif, *Nano-Ionic Redox Resistive RAM–Device Performance Enhancement Through Materials Engineering, Characterization and Electrical Testing*, Ph.D. thesis (2014).
[84] C. Gopalan, Y. Ma, T. Gallo, J. Wang, E. Runnion, J. Saenz, F. Koushan, P. Blanchard, and S. Hollmer, Solid-State Electronics **58**, 54 (2011).
[85] "Breakthrough resistive ram (reram) technology,".
[86] M. Lanza, Materials **7**, 2155 (2014).
[87] X. Hong, D. J. Loy, P. A. Dananjaya, F. Tan, C. Ng, and W. Lew, Journal of materials science **53**, 8720 (2018).
[88] L. Goux, A. Fantini, A. Redolfi, C. Chen, F. Shi, R. Degraeve, Y. Y. Chen, T. Witters, G. Groeseneken, and M. Jurczak, in *2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers* (IEEE, 2014) pp. 1–2.
[89] S. Clima, K. Sankaran, Y. Y. Chen, A. Fantini, U. Celano, A. Belmonte, L. Zhang, L. Goux, B. Govoreanu, R. Degraeve, *et al.*, physica status solidi (RRL)–Rapid Research Letters **8**, 501 (2014).
[90] D. J. Wouters, in *43rd IEEE Semiconductor Interface Specialists Conference, San Diego* (2012).
[91] K. Moon, A. Fumarola, S. Sidler, J. Jang, P. Narayanan, R. M. Shelby, G. W. Burr, and H. Hwang, IEEE Journal of the Electron Devices Society **6**, 146 (2017).
[92] A. Fumarola, S. Sidler, K. Moon, J. Jang, R. M. Shelby, P. Narayanan, Y. Leblebici, H. Hwang, and G. W. Burr, IEEE Journal of the Electron Devices Society **6**, 169 (2017).
[93] M. Latif, T. Nichol, M. Mitkova, D. Tenne, I. Csarnovics, S. Kokenyesi, and A. Csik, in *2014 IEEE Workshop On Microelectronics And Electron Devices (WMED)* (IEEE, 2014) pp. 1–4.
[94] M. Latif, I. Csarnovics, S. Kökényesi, A. Csik, and M. Mitkova, Canadian Journal of Physics **92**, 623 (2013).
[95] A. Grossi, E. Nowak, C. Zambelli, C. Pellissier, S. Bernasconi, G. Cibrario, K. El Hajjam, R. Crochemore, J. F. Nodin, P. Olivo, and L. Perniola, in *2016 IEEE International Electron Devices Meeting (IEDM)* (IEEE, 2016) pp. 4–7.
[96] M. Azzaz, A. Benoist, E. Vianello, D. Garbin, E. Jalaguier, C. Cagli, C. Charpin, S. Bernasconi, S. Jeannot, T. Dewolf, G. Audoit, C. Guedj, S. Denorme1, P. Candelier1, C. Fenouillet-Beranger, and L. Perniola, in *2015 45th European Solid State Device Research Conference (ESSDERC)* (IEEE, 2015) pp. 266–269.
[97] B. Govoreanu, G. Kar, Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. Radu, L. Goux, S. Clima, R. Degraeve, *et al.*, in *IEEE International Electron Devices Meeting (IEDM)* (IEEE, 2011) pp. 31–6.
[98] S. Clima, Y. Chen, A. Fantini, L. Goux, R. Degraeve, B. Govoreanu, G. Pourtois, and M. Jurczak, IEEE Electron Device Letters **36**, 769 (2015).
[99] Y. Y. Chen, R. Degraeve, S. Clima, B. Govoreanu, L. Goux, A. Fantini, G. S. Kar, G. Pourtois, G. Groeseneken, D. J. Wouters, *et al.*, in *IEEE International Electron Devices Meeting (IEDM)* (2012) pp. 20–3.
[100] "Cmp-leti memory advanced demonstrator 200mm (mad200),".
[101] P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S. P. Wong, *et al.*, Nature communications **8**, 1 (2017).
[102] P. Kumbhare, I. Chakraborty, A. Khanna, and U. Ganguly, IEEE Transactions on Electron Devices **64**, 3967 (2017).
[103] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, Nanotechnology **23**, 075201 (2012).
[104] W. He, H. Sun, Y. Zhou, K. Lu, K. Xue, and X. Miao, Scientific Reports **7**, 10070 (2017).
[105] W. Wu, H. Wu, B. Gao, N. Deng, S. Yu, and H. Qian, IEEE Electron Device Letters **38**, 1019 (2017).
[106] S. Yu, D. Kuzum, and H.-S. P. Wong, in *IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE, 2014) pp. 1062–1065.
[107] F. Zahoor, T. Z. Azni Zulkifli, and F. A. Khanday, Nanoscale Research Letters **15**, 1 (2020).
[108] M. Zhao, B. Gao, J. Tang, H. Qian, and H. Wu, Applied Physics Reviews **7**, 011301 (2020).
[109] K. Beckmann, J. Holt, H. Manem, J. Van Nostrand, and N. C. Cady, MRS Advances **1**, 3355 (2016).
[110] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, *et al.*, Nature Electronics **1**, 52 (2018).
[111] M. Zhao, B. Gao, Y. Xi, F. Xu, H. Wu, and H. Qian, IEEE Journal of the Electron Devices Society **7**, 1239 (2019).
[112] P.-Y. Chen and S. Yu, in *2018 IEEE International Reliability Physics Symposium (IRPS)* (IEEE, 2018) pp. 5C–4.
[113] U. Russo, D. Kamalanathan, D. Ielmini, A. L. Lacaita, and M. N. Kozicki, IEEE transactions on electron devices **56**, 1040 (2009).
[114] C.-X. Xue, T.-Y. Huang, J.-S. Liu, T.-W. Chang, H.-Y. Kao, J.-H. Wang, T.-W. Liu, S.-Y. Wei, S.-P. Huang, W.-C. Wei, *et al.*, in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)* (IEEE, 2020) pp. 244–246.
[115] A. Valentian, F. Rummens, E. Vianello, T. Mesquida, C. L.-M. de Boissac, O. Bichler, and C. Reita, in *2019 IEEE International Electron Devices Meeting (IEDM)* (IEEE, 2019) pp. 14–3.
[116] F. Alibart, E. Zamanidoost, and D. B. Strukov, Nature communications **4** (2013).
[117] T. Chang, Y. Yang, and W. Lu, IEEE Circuits and Systems Magazine **13**, 56 (2013).
[118] T. Serrano-Gotarredona and B. Linares-Barranco, in *Electronics, Circuits and Systems (ICECS), 2012 19th IEEE International Conference on* (IEEE, 2012) pp. 949–952.
[119] G. Indiveri, R. Legenstein, G. Deligeorgis, and T. Prodromakis, Nanotechnology **24**, 384010 (2013).
[120] S. Saïghi, C. G. Mayr, T. Serrano-Gotarredona, H. Schmidt, G. Lecerf, J. Tomas, J. Grollier, S. Boyn, A. F. Vincent, D. Querlioz, S. La Barbera, F. Alibart, D. Vuillaume, O. Bichler, C. Gamrat, and B. Linares-Barranco, Frontiers in neuroscience **9** (2015).
[121] V. Saxena, X. Wu, I. Srivastava, and K. Zhu, in *Proceedings of the 4th ACM International Conference on Nanoscale Computing and Communication* (ACM, 2017) p. 18.
[122] M. Prezioso, M. Mahmoodi, F. M. Bayat, H. Nili, H. Kim, A. Vincent, and D. Strukov, Nature communications **9**, 1 (2018).
[123] C. Walczyk, D. Walczyk, T. Schroeder, T. Bertaud, M. Sowinska, M. Lukosius, M. Fraschke, D. Wolansky, B. Tillack, E. Miranda, *et al.*, IEEE transactions on electron devices **58**, 3124 (2011).
[124] V. Saxena, in *IEEE International Symposium on Circuits & Systems (ISCAS)* (2020).
[125] V. Saxena, in *(invited) IEEE Int. Midwest Symposium on Circuits and Systems (MWSCAS)* (2019).
[126] X. Wu, V. Saxena, and K. A. Campbell, in *SPIE Sensing Technology+ Applications* (International Society for Optics and Photonics, 2014) pp. 911906–911906.
[127] V. Saxena, in *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)* (2018) pp. 1–5.
[128] S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, N. Ramaswamy, and D. Ielmini, in *2016 IEEE Symposium on VLSI Technology* (IEEE, 2016) pp. 1–2.
[129] Z. Wang, S. Joshi, S. E. Savelev, H. Jiang, R. Midya, P. Lin, M. Hu, N. Ge, J. P. Strachan, Z. Li, Z. Li, Q. Wu, M. Barnell, G.-L. Li, L. Xin, Huolin, R. S. Williams, Q. Xia, and J. J. Yang, Nature materials **16**, 101 (2017).
[130] C. Du, W. Ma, T. Chang, P. Sheridan, and W. D. Lu, Advanced Functional Materials **25**, 4290 (2015).
[131] T. Hirtzlin, M. Bocquet, B. Penkovsky, J.-O. Klein, E. Nowak, E. Vianello, J.-M. Portal, and D. Querlioz, Frontiers in Neuroscience **13** (2019).
[132] V. Joshi, M. Le Gallo, S. Haefeli, I. Boybat, S. R. Nandakumar, C. Piveteau, M. Dazzi, B. Rajendran, A. Sebastian, and E. Eleftheriou, Nature Communications **11**, 1 (2020).

[133]Q. Liu, B. Gao, P. Yao, D. Wu, J. Chen, Y. Pang, W. Zhang, Y. Liao, C.-X. Xue, W.-H. Chen, *et al.*, in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)* (IEEE, 2020) pp. 500–502.
[134]K. Kanda, N. Shibata, T. Hisada, K. Isobe, M. Sato, Y. Shimizu, T. Shimizu, T. Sugimoto, T. Kobayashi, N. Kanagawa, *et al.*, IEEE journal of solid-state circuits **48**, 159 (2012).
[135]G. H. Kim, H. Ju, M. K. Yang, D. K. Lee, J. W. Choi, J. H. Jang, S. G. Lee, I. S. Cha, B. K. Park, J. H. Han, *et al.*, Small **13**, 1701781 (2017).
[136]J.-C. Liu, T.-Y. Wu, and T.-H. Hou, IEEE Transactions on Circuits and Systems II: Express Briefs **65**, 617 (2018).
[137]W. Zhang, X. Peng, H. Wu, B. Gao, H. He, Y. Zhang, S. Yu, and H. Qian, in *2019 56th ACM/IEEE Design Automation Conference (DAC)* (IEEE, 2019) pp. 1–6.
[138]Z. Alamgir, K. Beckmann, J. Holt, and N. C. Cady, Applied Physics Letters **111**, 063111 (2017).
[139]J. M. Correll, V. Bothra, F. Cai, Y. Lim, S. H. Lee, S. Lee, W. D. Lu, Z. Zhang, and M. P. Flynn, **6** (2020), 10.1109/JXCDC.2020.2992228.
[140]J. M. Cruz-Albrecht, M. W. Yung, and N. Srinivasa, IEEE transactions on biomedical circuits and systems **6**, 246 (2012).
[141]X. Wu, V. Saxena, K. Zhu, and S. Balagopal, IEEE Transactions on Circuits and Systems II: Express Briefs **62**, 1088 (2015).
[142]B. D. Sahoo, in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on* (IEEE, 2017) pp. 1–4.
[143]B. Larras, P. Chollet, C. Lahuec, F. Seguin, and M. Arzel, in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE, 2017) pp. 1–4.
[144]I. Sourikopoulos, S. Hedayat, C. Loyez, F. Danneville, V. Hoel, E. Mercier, and A. Cappy, Frontiers in Neuroscience **11**, 123 (2017).
[145]A. Joubert, B. Belhadj, and R. Héliot, in *New Circuits and Systems Conference (NEWCAS), 2011 IEEE 9th International* (IEEE, 2011) pp. 9–12.
[146]W. Wan, R. Kubendran, S. B. Eryilmaz, W. Zhang, Y. Liao, D. Wu, S. Deiss, B. Gao, P. Raina, S. Joshi, *et al.*, in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)* (IEEE, 2020) pp. 498–500.
[147]I. E. Ebong and P. Mazumder, Proceedings of the IEEE **100**, 2050 (2012).
[148]X. Wu, V. Saxena, K. Zhu, and S. Balagopal, accepted in the IEEE Transactions on Circuits and Systems II: Express Briefs (2015), 10.1109/TCSII.2015.2456372.
[149]X. Wu, *Analog Spiking Neuromorphic Circuits and Systems for Brain- and Nanotechnology-Inspired Cognitive Computing*, Ph.D. thesis (2016).
[150]C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **32**, 1201 (2013).
[151]P. Y. Simard, D. Steinkraus, and J. C. Platt, in *null* (IEEE, 2003) p. 958.
[152]L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, in *International Conference on Machine Learning* (2013) pp. 1058–1066.
[153]A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Advances In Neural Information Processing Systems* (2012) pp. 1097–1105.
[154]K. He, X. Zhang, S. Ren, and J. Sun, in *European conference on computer vision* (Springer, 2016) pp. 630–645.
[155]K. Simonyan and A. Zisserman, arXiv preprint arXiv:1409.1556 (2014).
[156]P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, in *International Joint Conference onNeural Networks (IJCNN)* (2015) pp. 1–8.
[157]E. Hunsberger and C. Eliasmith, arXiv preprint arXiv:1611.05141 (2016).
[158]B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, Frontiers in neuroscience **11**, 682 (2017).
[159]A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, Frontiers in neuroscience **13**, 95 (2019).
[160]P. U. Diehl and M. Cook, Frontiers in Computational Neuroscience **9** (2015).
[161]S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, Neural Networks **99**, 56 (2018).
[162]R. Vaila, J. Chiasson, and V. Saxena, arXiv:1903.12272.
[163]E. O. Neftci, C. Augustine, S. Paul, and G. Detorakis, Frontiers in neuroscience **11**, 324 (2017).
[164]J. H. Lee, T. Delbruck, and M. Pfeiffer, Frontiers in Neuroscience **10** (2016).
[165]P. O'Connor, E. Gavves, and M. Welling, arXiv preprint arXiv:1706.04159 (2017).
[166]S. R. Kulkarni and B. Rajendran, Neural Networks **103**, 118 (2018).
[167]H. Mostafa, IEEE transactions on neural networks and learning systems **29**, 3227 (2017).
[168]S. B. Shrestha and G. Orchard, in *Advances in Neural Information Processing Systems* (2018) pp. 1412–1421.
[169]H. Mostafa, V. Ramesh, and G. Cauwenberghs, Frontiers in neuroscience **12**, 608 (2018).
[170]E. O. Neftci, H. Mostafa, and F. Zenke, IEEE Signal Processing Magazine **36**, 61 (2019).
[171]E. Hunsberger and C. Eliasmith, arXiv preprint arXiv:1510.08829 (2015).
[172]C. Lee, P. Panda, G. Srinivasan, and K. Roy, Frontiers in neuroscience **12**, 435 (2018).
[173]S. Nandakumar, I. Boybat, M. Le Gallo, E. Eleftheriou, A. Sebastian, and B. Rajendran, Scientific reports **10**, 1 (2020).
[174]T. Masquelier, R. Guyonneau, and S. J. Thorpe, PloS One **3**, e1377 (2008).
[175]T. Masquelier, R. Guyonneau, and S. J. Thorpe, Neural computation **21**, 1259 (2009).
[176]R. Vaila, J. Chiasson, and V. Saxena, arXiv preprint arXiv:1903.12272 (2019).
[177]A. Tavanaei and A. S. Maida, arXiv preprint arXiv:1611.03000 (2016).
[178]R. Vaila, J. Chiasson, and V. Saxena, arXiv preprint arXiv:2002.11843 (2020).
[179]R. Vaila, J. Chiasson, and V. Saxena, in *International Conference on Neuromorphic Systems (ICONS)* (2019).
[180]R. Vaila, J. Chiasson, and V. Saxena, arXiv preprint arXiv:2005.04167 (2020).
[181]S. M. Bohte, J. N. Kok, and H. La Poutre, Neurocomputing **48**, 17 (2002).
[182]M. Jaderberg, W. M. Czarnecki, S. Osindero, O. Vinyals, A. Graves, D. Silver, and K. Kavukcuoglu, in *International Conference on Machine Learning* (2017) pp. 1627–1635.
[183]T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, Nature communications **7**, 13276 (2016).
[184]N. Anwani and B. Rajendran, in *2015 international joint conference on neural networks (IJCNN)* (IEEE, 2015) pp. 1–8.
[185]F. Zenke and S. Ganguli, Neural computation **30**, 1514 (2018).
[186]S. Yu, Z. Li, P.-Y. Chen, H. Wu, B. Gao, D. Wang, W. Wu, and H. Qian, in *2016 IEEE International Electron Devices Meeting (IEDM)* (IEEE, 2016) pp. 16–2.
[187]X. Guo, F. M. Bayat, M. Bavandpour, M. Klachko, M. Mahmoodi, M. Prezioso, K. Likharev, and D. Strukov, in *2017 IEEE International Electron Devices Meeting (IEDM)* (IEEE, 2017) pp. 6–5.
[188]C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, *et al.*, Nature communications **9**, 1 (2018).
[189]Z. Wang, S. Joshi, S. Savelev, W. Song, R. Midya, Y. Li, M. Rao, P. Yan, S. Asapu, and Y. Zhuo, Nature Electronics **1**, 137 (2018).
[190]C.-X. Xue, W.-H. Chen, J.-S. Liu, J.-F. Li, W.-Y. Lin, W.-E. Lin, J.-H. Wang, W.-C. Wei, T.-W. Chang, T.-C. Chang, *et al.*, in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)* (IEEE, 2019) pp. 388–390.
[191]J. Schemmel, J. Fieres, and K. Meier, in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational*

Intelligence). IEEE International Joint Conference on (IEEE, 2008) pp. 431–438.
[192]J. Schemmel, D. Briiderle, A. Griibl, M. Hock, K. Meier, and S. Millner, in *Circuits and systems (ISCAS), proceedings of 2010 IEEE international symposium on* (IEEE, 2010) pp. 1947–1950.
[193]T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. A. Petrovici, M. Schmuker, D. Brüderle, J. Schemmel, and K. Meier, Frontiers in neuroscience **7** (2013).
[194]E. Painkras, L. Plana, J. Garside, S. Temple, S. Davidson, J. Pepper, D. Clark, C. Patterson, and S. Furber, in *Custom Integrated Circuits Conference (CICC), 2012 IEEE* (IEEE, 2012) pp. 1–4.
[195]C. Mayr, S. Hoeppner, and S. Furber, arXiv preprint arXiv:1911.02385 (2019).
[196]C. Liu, G. Bellec, B. Vogginger, D. Kappel, J. Partzsch, F. Neumärker, S. Höppner, W. Maass, S. B. Furber, R. Legenstein, *et al.*, Frontiers in neuroscience **12**, 840 (2018).
[197]D. Garbin, E. Vianello, O. Bichler, Q. Rafhay, C. Gamrat, G. Ghibaudo, B. DeSalvo, and L. Perniola, IEEE Transactions on Electron Devices **62**, 2494 (2015).
[198]K.-H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, and W. Lu, Nano letters **12**, 389 (2011).
[199]M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, Nature **521**, 61 (2015).
[200]S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z.-Q. Wang, A. Calderoni, N. Ramaswamy, and D. Ielmini, IEEE Transactions on Electron Devices **63**, 1508 (2016).