A Hardware Architecture for Columnar-Organized Memory Based on CMOS Neuron and Memristor Crossbar Arrays

Jafar Shamsi, Karim Mohammadi, and Shahriar B. Shokouhi, *Member, IEEE*

***Abstract*— Neuromorphic utilizes VLSI technology to implement a brain-inspired architecture. Recently, a brain-inspired associative memory with large capacity and robust retrieval, known as columnar-organized memory (COM), has been introduced. COM is a combination of spiking winner-take-all (WTA) units, which is inspired by the cortex structure. In this paper, a hardware architecture of COM is proposed that is presented at three levels of design. At the level I, a low-power circuit of a leaky integrate and fire neuron is introduced that is compatible with the architecture of COM. At the level II, the assembly of the proposed neuron and a single memristor crossbar array are used to implement a WTA module. At the level III, a COM hardware architecture is developed using the combination of the WTA modules and memristor crossbar arrays. The ex situ method is utilized to train the COM hardware. The simulations are performed at all design levels. First, the power consumption of the neuron circuit is evaluated. It consumes 4.3 pJ/spike, and its static power is 182 pW. Second, the operation of the WTA module is deliberated. Finally, the operation of the COM hardware for message storage and retrieval is evaluated in the presence of the hardware imperfections.**

***Index Terms*— Columnar-organized memory (COM), memristor crossbar, neuron circuit, synaptic circuit, winner take all (WTA).**

I. Introduction

NEUROMORPHIC is an approach for hardware implementation of a brain-inspired system to benefit from brain capabilities. Brain is an energy-efficient system to perform complex cognitive tasks such as vision and memory. To archive the efficiency, the brain uses several techniques such as sparse coding, analog computing, and communicating through low amplitude spikes [1]. Moreover, highly parallel architecture and plasticity allow it to achieve the robustness to the component failure and noises [1]. In this regard, neuromorphic utilizes the brain-inspired architectures and VLSI technology to investigate an energy-efficient and robust brain-inspired system.

A main cognitive task of the brain is associative memory that stores data by linking them to each other and retrieves data robustly. In the literature, several architectures for neural associative memory have been proposed such as Hopfield neural network (HNN) [2], bidirectional associative memory (BAM) [3], clique-based neural network (CBNN) [4], and columnar-organized memory (COM) [5]. HNN and BAM are classic associative memory based on the binary neurons that suffer from the low storage capacity [5]. Although CBNN is capable to store a large number of messages, it utilizes binary synapses and neurons to store binary data. Recently, a spiking associative memory with large capacity and robust retrieval has been introduced and compared with HNN, BAM, and CBNN that is called the COM [5]. Its architecture is inspired by the brain cortex structure. It comprises N spiking winner-take-all (WTA) networks ($N > 1$). A spiking WTA comprises n spiking neurons, which the most stimulated neuron is activated (winner) to elicit spikes. The WTA operation fulfills the sparse coding, where a small fraction of neurons is activated simultaneously in the entire of COM. The sparse coding enables COM to store a large number of data. In addition, the spiking WTAs are linked to each other through the lateral excitatory synapses that results in robust message retrieval. These features make the COM appropriate for utilization in a neuromorphic system.

In this paper, a new hardware architecture is proposed to implement the COM. The implementation is accomplished at three levels of hardware design. At the lowest level (level I), the circuit design of the basic components is considered with respect to the power and area cost. The basic components are spiking neuron and synaptic circuits. A considerable amount of literature has been published on the design of the spiking neuron and synaptic circuits that are summarized as follows.

A. Spiking Neuron Circuit

Spiking neurons are distributed processing units which are described through a large range of models [6]. However, in a neuromorphic design, the leaky integrate and fire (LIF) model is usually used to implement the spiking neuron circuit. The axon hillock is the first circuit that is introduced by Mead [7] in the late 1980s. The axon hillock is very compact; however, it dissipates significant power. In addition, it lacks biological features such as refractory period and adaptive spike frequency. Furthermore, its firing threshold voltage depends on CMOS parameters. An alternative neuron circuit with the refractory period and adjustable firing threshold is introduced

Manuscript received August 30, 2017; revised January 5, 2018; accepted February 24, 2018. Date of publication April 3, 2018; date of current version November 30, 2018. This work was supported by the Iran Neural Technology Centre, Iran University of Science and Technology, Tehran, Iran. *(Corresponding author: Jafar Shamsi.)*

The authors are with the School of Electrical Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran (e-mail: Jafarshamsi@elec.iust.ac.ir; mohammadi@iust.ac.ir; bshokouhi@iust.ac.ir).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2018.2815025

in [8]. Similar to the axon hillock, it dissipates much power due to its output buffer. In [9], a compact LIF neuron is introduced which is optimized for power consumption. It also benefits biological features such as refractory period and adjustable firing threshold. In spite of the fact that its static power is nil, it still consumes much energy per spike (900 pJ). In [10], a low-power LIF neuron is proposed with power consumption of 0.4 pJ/spike. This neuron is optimized to drive an active synaptic circuit where the synaptic circuit has no sink current. In addition, it requires an output buffer to drive a large load. In order to drive large resistive loads, a dual mode LIF neuron is presented in [11]. The neuron operates in a dual mode for spike integration and firing. When it elicits spikes, it is capable of driving large resistive synapses. It also contains WTA bus interface that is used to implement the lateral inhibition. The power consumption of the neuron is 9.6 pJ/spike/synapse. Although this power consumption is low, it consumes permanent static power. In [12], a low-power LIF neuron has been introduced which consumes about 2 pJ/spike/synapse and provides a global reset connection to implement lateral inhibition in a spiking WTA.

B. Synaptic Circuits

Synapses are distributed memory elements in neural networks. Hence, the adjustable and nonvolatile storage of synaptic weights are key features of a synaptic circuit [13]. There are several circuit elements to implement the synapses such as resistor, capacitor/transistors, floating-gate transistors, and memristor. The resistor's static behavior makes it impossible to be adjusted; therefore, it is not applicable in a trainable circuit [14]. Combination of capacitor and CMOS transistors is another solution to implement synaptic circuits [15]. When a MOSFET transistor is biased in the subthreshold region, its current is exponentially related to the voltage difference between the gate and source voltage $V_{\rm gs}$ [16], [17]. In other words, the synaptic weight is related to $V_{\rm gs}$ that can be stored in a capacitor. However, a capacitor cannot store charges (i.e., synaptic weight) for long periods of time because of its permanent charge leakage [16]. A floating-gate transistor is a successful alternative for working as synapses to store the synaptic weights in a nonvolatile manner. Hot-electron injection and Fowler–Northeim tunneling are used to change stored charge on the floating gate and adjust the synaptic weight [18]. Recently, emerging memristor as a nonvolatile analog memory opens a new research area for synaptic circuit implementation [19]. Memristor is a nanoscale two-port passive element which is appropriate for implementing a compact synaptic circuit. Specifically, memristor crossbar is a low-power, scalable, and compact array as a promising candidate to implement the synaptic circuit in a neuromorphic chip [20], [21]. The memristor crossbar array has been used in several brain-inspired architectures such as hyperdimensional associative memory [22], brain-state-in-a-box associative memory [23], Hopfield associative memory [24], and spiking WTA [11].

We propose a low-power circuit of LIF neuron at the design level I that is compatible with the COM architecture. It utilizes three terminals for feedforward synapses, lateral inhibitory connections, and lateral excitatory synapses. In addition, the memristor crossbar array is utilized to implement the synaptic circuits. At the design level II, the proposed neuron and synaptic circuit are combined to implement a WTA module. The proposed neuron is used at the input and output layers of the WTA module. Furthermore, the memristor crossbar arrays are employed to implement the feedforward synaptic circuit between the input and output layers of the WTA module. At the design level III, the WTA modules are combined with the memristor crossbar arrays to implement a COM architecture. Training of the COM hardware is performed using *ex situ* method. In this regard, the spike timing-dependent plasticity (STDP) and Hebbian rules have been utilized to calculate the synaptic weights. The final synaptic weights are programmed to the memristor crossbar arrays accordingly.

In order to evaluate the proposed architecture, the simulation is performed at all levels of design. First, the operation of the proposed neuron is simulated. In addition, its power consumption is evaluated and compared with the related works. Second, the operation of the WTA module is simulated for two pattern sets. Finally, the robustness and capacity of the COM hardware are evaluated. In this regard, the random messages are stored in the COM hardware. Then, the noisy and erased messages are used for message retrieval. In addition, the hardware imperfections such as stuck-at-faults (SAFs) and process variations are considered for the memristor crossbar arrays. The simulation results demonstrate the perfect operation of the proposed architecture at all design levels.

Rest of this paper is organized in the following way. In Section II, the architecture of COM and its operation for message storage and retrieval are reviewed. The hardware implementation of COM at three design levels is presented in Section III. Simulation results including the power consumption and operation of the proposed neuron, WTA, and COM are provided in Section IV. The concluding remarks are discussed in Section V.

II. Columnar-Organized Memory (COM): Architecture and Operation

Details of the COM have been introduced in [5]. However, the architecture of COM, message storage, and message retrieval is reviewed here. The architecture of COM is shown in Fig. 1. It comprises of N spiking WTAs with lateral excitatory synapses between them. A WTA consists of an input layer, an output layer, and lateral inhibitory connections between the output layer neurons. The input layer of a WTA consists of l LIF neurons. The input neurons are fully connected to the output layer through the feedforward synapses. Output layer is an assembly of n LIF neurons. These neurons are linked to each other through the lateral inhibitory connections. There are also lateral excitatory synapses that link a neuron of a WTA with neurons of other WTAs. The weights of the feedforward synapses between the input and output layers of a WTA and the lateral excitatory synapses are adjusted through the message storage. In Sections II-A and II-B, message storage and retrieval are reviewed.

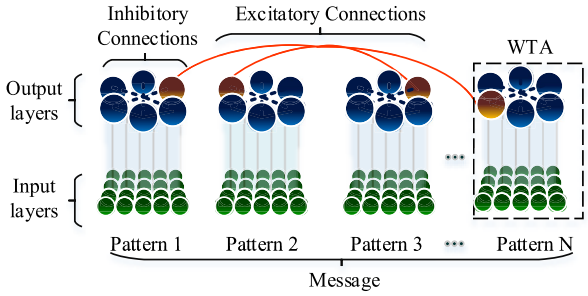

Fig. 1. COM architecture comprising the WTAs with lateral excitatory synapses between them.

A. Message Storage

Message M is a vector of N patterns ($M = \{p_1, p_2, p_3, \ldots, p_N\}$). Pattern p is a vector of length l with binary or real-value elements ranged from 0 to 1 ($p = [x_1, x_2, x_3, \ldots, x_l]$). There is a relation between the architecture of a COM and the structure of a message. The number of patterns N in a message is equal to the number of the WTAs in a COM. (Pattern p_i is corresponding to the ith WTA that is denoted by WTA$_i$.) Moreover, the length of pattern l is equal to the number of input neurons in a WTA (see Fig. 1).

Message storage of COM contains two steps; pattern storage and pattern association. At the pattern storage step, N pattern sets are used to train N WTAs. Fig. 2(a) shows three pattern sets (P_1, P_2, P_3) and corresponding WTAs (WTA$_1$, WTA$_2$, WTA$_3$). A pattern p_k from a pattern set P_i is applied to train the neuron n_k of the WTA$_i$. For instance, the pattern $p_1 = A$ of the pattern set P_1 is used to train the neuron n_1 of the WTA$_1$ [see Fig. 2(a)]. In this regard, the synaptic weights of a neuron n_i are adjusted through a training algorithm. STDP rule is usually applied to adjust the synaptic weights in a spiking WTA [5], [11], [25]–[31]. The weight change Δw depends on the differences between pre- and postsynaptic spike time [32], [33]. The basic model of the STDP rule is determined by

$$\Delta w = \begin{cases} A^+ e^{-\Delta t/\tau^+}, & \text{if } \Delta t > 0 \\ -A^- e^{\Delta t/\tau^-}, & \text{if } \Delta t < 0 \end{cases} \tag{1}$$

where Δt is the time differences between presynaptic and postsynaptic spikes ($\Delta t = t_{\text{post}} - t_{\text{pre}}$). Parameters of A^+ and A^- are the maximum and minimum value of Δw, and τ^+ and τ^- are constants. Using the STDP rule, a pattern p_i is linked to an output neuron. The final weights of the neuron are similar to the corresponding pattern [5]. This similarity between the pattern and synaptic weights is analogous to store the patterns in the WTA.

At the pattern association step, M messages are stored in COM by adjusting the lateral excitatory synapses. In order to store a message $M = \{p_1, p_2, p_3, \ldots, p_N\}$, a neuron corresponding to pattern $p_i \in M$ is linked to a neuron corresponding to pattern $p_{k \neq i} \in M$ through the lateral excitatory synapses. Fig. 2(b) displays two messages (m_1 and m_2) that are stored in COM. For instance, the neurons that are corresponding to the patterns of message m_1 are linked to each other through the lateral excitatory synapses (red solid curves). Also, message m_2 has been stored accordingly. Weights of the excitatory synapses are determined by

$$w_{\left(n_{ij}, n'_{ij}\right)} = \begin{cases} w^{\text{exc}} & \text{if } n_{ij} \equiv p_i,\ n'_{ij} \equiv p'_i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where n_{ij} represents the jth neuron in the WTA$_i$. The term $n_{ij} \equiv p_i$ determines that the jth neurons in the WTA$_i$ is corresponding to the pattern p_i. The parameter of w^{exc} is a constant, where $0 \leq w^{\text{exc}} \leq 1$.

When a message is stored in COM, a clique is generated. A clique is an assembly of N neurons, in which each neuron belongs to a WTA. These neurons are fully connected to each other. When several messages are stored, the generated cliques may share common neurons. Fig. 2(b) illustrates two cliques for two sets of neurons: $c_1 = \{A, b, \delta\}$ and $c_2 = \{C, a, \beta\}$.

B. Message Retrieval

Message retrieval refers to the reactivation of a clique to represent a stored message m_i. When a given message m' is applied to COM, each pattern of the message m' is applied to the corresponding WTA (i.e., the pattern p'_i is applied to the WTA$_i$). Then, the neurons of the WTA$_i$ compete to win. The winner neuron elicits the highest spike rate in the entire population of the WTA$_i$. In addition, the neurons, which are associated with each other through the lateral excitatory synapses, cooperate to activate each other. In other words, neurons of a clique cooperate to activate each other. The winner neurons represent a stored message m which is the most similar to the applied message m'.

Fig. 2(c) shows the process of the message retrieval when a typical message m' is applied to COM. The elements of message m' are provided for the corresponding WTAs. (p_1, p_2, and p_3 are provided for WTA$_1$, WTA$_2$, and WTA$_3$, respectively.) For instance, the pattern $p_1 = A$ is delivered to the WTA$_1$. Then, the neurons of the WTA$_1$ compete to elicit spikes. The most stimulated neuron (neuron A) is activated and elicits spikes. The activated neuron A sends the spikes to other neurons of the clique $c_1 = \{A, b, \delta\}$ through lateral excitatory connections. It means that neuron A transmits the spikes to neuron b and δ. This process will be performed in all of the WTAs. The winner neurons (A, b, and δ) represent a message (m_1), which is the most similar one to the message m'.

III. COM Hardware Architecture

In this section, we explain the hardware implementation of COM at three levels of design. At the design level I, a low-power circuit of an LIF neuron is proposed. In addition, the synaptic circuit is designed using a memristor crossbar array. The hardware problems of the memristor crossbar, such as process variations, SAFs, IR drop, and sneak paths, are discussed in this section as well. At the design level II, the proposed neuron and synaptic circuits are assembled to implement the spiking WTA. The proposed neuron is used in the input layer and output layer of the WTAs. The memristor crossbar array is also provided to implement the feedforward

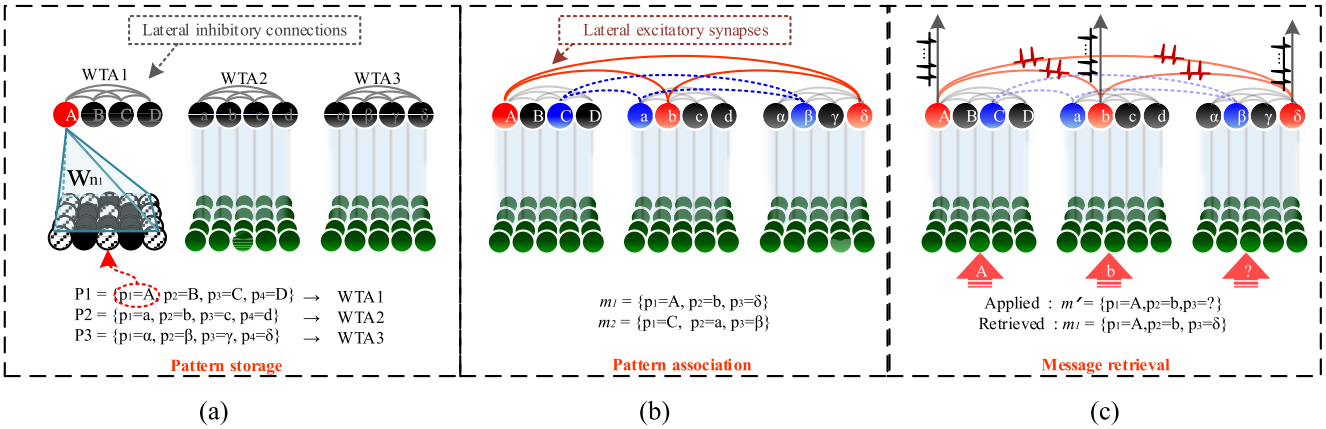

Fig. 2. Message storage and retrieval in a COM architecture with three WTAs. (a) Three pattern sets are applied to train the corresponding WTAs. (b). Two messages m_1 and m_2 are stored in the COM by adjusting the lateral excitatory connections. (c) Retrieval of message m_1, when a given pattern m' is applied to the COM.

synaptic circuit between the input layer and output layer of the WTA. At the design level III, the assembly of WTAs and memristor crossbar arrays are utilized to implement a COM hardware. Finally, the *ex situ* training method is applied to program the synaptic circuits (memristor crossbar arrays) of the COM hardware.

A. Design Level I: Neuron and Synaptic Circuit

At this level, a spiking neuron and synaptic circuit are designed.

1) LIF Neuron Circuit: The circuit of an LIF neuron comprises several sections such as current integration, leakage, spike generation, membrane potential reset, and refractory period. Fig. 3 shows the proposed LIF neuron circuit which includes capacitor C_u for current integration, transistor $M5$ for leakage, Schmitt trigger for spike generation, $M11$–$M14$ for reset, and capacitor C_{ref} for refractory period. Terminals T_{FF} and T_{EX} are used to apply an input voltage to the neuron. For instance, the input voltages can be provided from the feedforward synapses (V_{FF}) and the lateral excitatory synapses (V_{EX}). The lateral inhibitory terminal T_{INH} is used to reset the membrane potential. The circuit has been designed with a 90-nm CMOS technology. The parameters of the circuit are shown in Table I. Parameters of $M1$ and $M2$ controls the range of the input current and the parameter of $M4$ controls the injected current into C_u. In addition, the capacitance of C_u and C_{ref} can be altered to control the output spikes rate.

When the input voltage $V_{\text{FF}}/V_{\text{EX}}$ is applied to the neuron, the current I_{in} is injected into the leaky integrator section through the current mirror. Then, the injected current is integrated by C_u and is leaked through $M5$. A low-power Schmitt trigger is developed to generate spike [34]. When the membrane potential reaches the switching voltage of the Schmitt trigger, the node *Rst* changes from high to low. Simultaneously, transistor $M11$ is turned ON and the membrane potential resets to zero through the reset section. Consequently, a spike is generated at the output of the neuron. In addition, it is possible to reset the membrane potential through the lateral inhibitory terminal T_{INH} which is used to implement lateral inhibition in

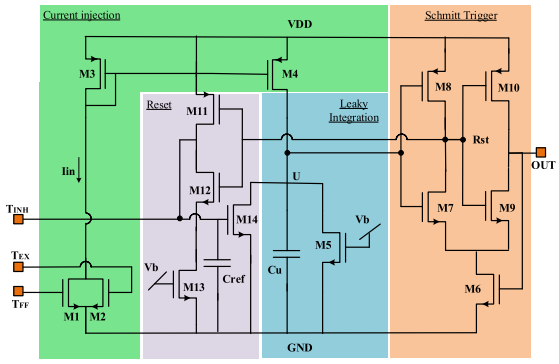

Fig. 3. Proposed circuit for the LIF neuron. It comprises several sections for current integration, leakage, spike generation, reset, and refractory period.

TABLE I
DESIGN PARAMETERS FOR THE PROPOSED NEURON CIRCUIT

Parameters	Value	Parameters	Value
W_{M10}	0.5~1 μm,	W_{M5}	0.4 μm
W_{M13}	1 μm	$L_{Mi\neq13}$	0.2 μm
W_{M1}, W_{M3}	0.12 μm	L_{M13}	0.1 μm
W_{M4}	0.12~0.2 μm	C_u	0.7~1.2 pF
W_{M2}	0.3~0.5 μm	C_{ref}	0.05 pF
W_{M6}, W_{M7}, W_{M8}, W_{M9}, W_{M11}, W_{M12}, W_{M14},	0.2 μm	vb	0.13 v

the spiking WTA. The switching voltage of the Schmitt trigger is the firing threshold of the neuron [34] and can be calculated by

$$V_{\text{SV}} = V_{\text{DD}}\frac{(R_{\text{nn}}-1)}{R_{\text{nn}}(R_{\text{np}}+1)+1} + V_{\text{th}}\frac{R_{\text{nn}}(2R_{\text{np}}-1)-1}{R_{\text{nn}}(R_{\text{np}}+1)+1} \tag{3}$$

where $R_{\text{np}} = (\beta_{M7/\beta_{M8}})^{1/2}$, $R_{\text{nn}} = (\beta_{M6/\beta_{M7}})^{1/2}$, and β_{Mi} corresponds to the transconductance of transistor Mi. The parameter of V_{th} is the threshold voltage of the transistors.

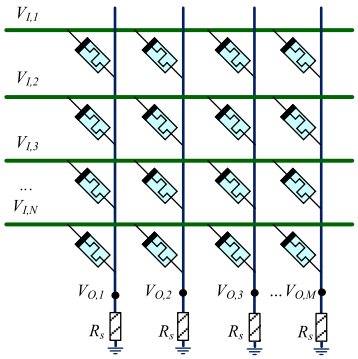

Fig. 4. Memristor crossbar array of $N \times M$. The input voltage V_I is applied to the WLs and the corresponding output voltage V_O is appeared in the BLs.

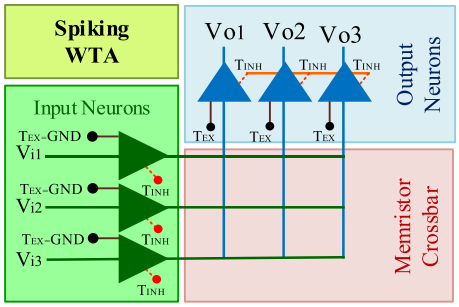

Fig. 5. Hardware architecture of a COM with three WTAs. The memristor crossbar arrays are used to implement both feedforward and lateral excitatory synapses.

When a biological neuron generates a spike, it is not capable of generating another spike during the refractory period. At the proposed neuron, capacitor C_{ref} models the refractory period time. When a spike is generated, the node T_{INH} is set to V_{DD}. Consequently, capacitor C_{ref} is charged toV_{DD} immediately. Then, it is discharged through $M13$ with the time constant $T = R_{M13}.C_{\text{ref}}$, where R_{M13} is the resistance of $M13$. The duration of the refractory period is related to the time constant T.

2) Synaptic Circuit: The memristor crossbar array has been used as the synaptic circuits for the both feedforward and the lateral excitatory synapses. Fig. 4 shows a typical $N \times M$ memristor crossbar array where an input voltage $V_I = [V_{I,1}, V_{I,2}V_{I,3}, \ldots, V_{I,N}]$ is applied to the word lines (WLs) and the current is collected through the bit lines (BLs). The output voltage $V_O = [V_{O,1}, V_{O,2}V_{O,3}, \ldots, V_{O,M}]$ is determined by $(R_s \to \infty)$

$$\begin{bmatrix} V_{O,1} \\ V_{O,2} \\ \cdots \\ V_{O,M} \end{bmatrix} = \begin{bmatrix} \frac{g_{1,1}}{N} & \frac{g_{1,2}}{N} & \cdots & \frac{g_{1,N}}{N} \\ \sum_{i=1} g_{i,1} & \sum_{i=1} g_{i,2} & & \sum_{i=1} g_{i,N} \\ \frac{g_{2,1}}{N} & \frac{g_{2,2}}{N} & \cdots & \frac{g_{2,N}}{N} \\ \sum_{i=1} g_{i,1} & \sum_{i=1} g_{i,2} & & \sum_{i=1} g_{i,N} \\ & & \cdots & \\ \frac{g_{M,1}}{N} & \frac{g_{M,2}}{N} & \cdots & \frac{g_{M,N}}{N} \\ \sum_{i=1} g_{i,1} & \sum_{i=1} g_{i,2} & & \sum_{i=1} g_{i,N} \end{bmatrix} \begin{bmatrix} V_{I,1} \\ V_{I,2} \\ \cdots \\ V_{I,N} \end{bmatrix} \quad (4)$$

where $g_{i,j}$ is the conductance of the memristor sitting on the connection between WL_i and BL_j.

In order to map a binary weight matrix to the conductance of a memristor crossbar array, the following relation is considered:

$$g_{i,j} = \begin{cases} g_{\text{Max}}, & w_{i,j} = w_{\text{Max}} \\ g_{\text{min}}, & w_{i,j} = w_{\text{min}} \end{cases} \quad (5)$$

where g_{Max} and g_{min} are the conductance of a memristor at low resistance state (LRS) and high resistance state (HRS), respectively.

Although the memristor crossbar array is a promising candidate for neuromorphic computing, the memristor crossbar suffers from several drawbacks such as process variation [20], [35]–[43], SAFs [44], sneak paths [45], and IR drop [38], [46]. The process variation and SAFs are caused due to the immature fabrication technology. An accurate control of the fabrication process or design techniques at the device level is necessary to overcome the issues (e.g., multilayer structure of memristor [40]). However, several software-based and hardware-based solutions have been introduced to resolve the problems [30], [37], [38], [42]–[44]. The hardware-based solutions increase the area overhead and power consumption. In addition, the software-based solutions usually involve being aware of the memristor crossbar imperfections to compensate problems. In this paper, the fault-tolerant design is achieved at the architecture level. In fact, the COM architecture is inherently fault tolerant to the SAFs and process variation.

Sneak paths are undesirable paths for current that are parallel to the intended path. In a spiking neuromorphic system based on a memristor crossbar array, the spikes are transmitted to a desirable neuron and other neurons (due to the sneak path) as well. However, the amplitude of the received spikes at the other neurons is degraded. Thus, these degraded spikes perform as noises. The COM architecture is also robust against such undesirable signals and noises.

IR drop is another issue that relies on the interconnect resistances. The resistance of the connections in a memristor crossbar causes the IR drop that degrades the applied voltage reaching the memristors. However, the reliability of reading and writing in a memristor crossbar array will be degraded if the size of memristor crossbar is beyond of 64×64 [46], [47]. In this paper, the effect of IR drop has been neglected.

B. Design Level II: WTA Circuit

At this level, a spiking WTA module is designed using the proposed neuron and memristor crossbar array. Fig. 5 demonstrates the hardware architecture of a spiking WTA module. It is constructed by three sections: input layer, synaptic circuit, and output layer. The input layer contains encoding neurons to encode the input patterns to the spike trains. The proposed neuron is used at the input layer to encode the injected current into the spike train. The rate of the spike trains is proportional to the input voltage. When the neuron is used in the input layer, the voltage is applied to the terminal T_{FF} and the output spikes are generated from the terminal OUT. Terminal T_{EX} is grounded, and terminal T_{INH} is left float (see Fig. 3).

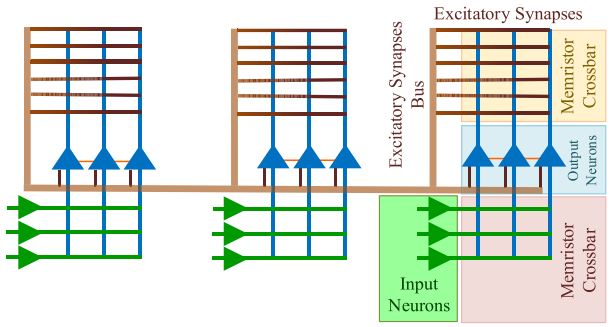

Fig. 6. Hardware architecture of a COM with three WTAs. The memristor crossbar arrays are used to implement both feedforward and lateral excitatory synapses.

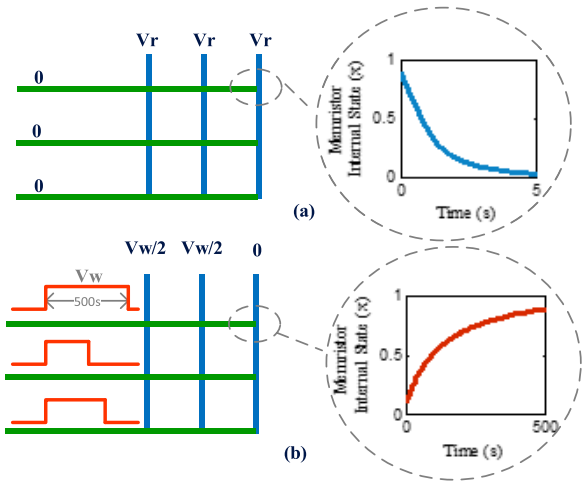

Fig. 7. Programming of a memristor crossbar array. (a) Resetting the memristance to HRS by applying a −2 V pulse. (b) Writing the synaptic weights in a column by applying a 2 V pulse. The highlighted curves demonstrate changing of the internal state of the memristor on the basis of the memristor model in [53] with the following parameters: $Vp = 1.5$ V, $Vn = 0.5$ V, $Ap = 0.005$, An $= 0.08$, $xp = 0.2$, $xn = 0.5$, $\alpha p = 1.2$, $\alpha n = 3$, $a1 = 3.7(10-7)$, $a2 = 4.35(10-7)$, $b = 0.7$, $x0 = 0.1$, $\eta = 1$.

The neurons of the output layer are also implemented using the proposed neuron circuits. When the proposed neuron is used in the output layer, the feedforward synapses are connected to terminal T_{FF}. In addition, all of the neurons are connected to each other through the terminal T_{INH} to implement the lateral inhibition. When an output neuron generates a spike, the membrane voltage of other neurons is reset to zero through terminal T_{INH} (see Fig. 3). While a neuron elicits spikes, other neurons are inhibited to generate spikes. The neuron with the highest spike rate is the winner and represents the applied pattern.

The feedforward synaptic circuit between the input and output layers is implemented using the memristor crossbar array. The crossbar rows receive the spike trains from the input layer and multiply them to the synaptic weights (memristance). The spikes are weighted corresponding to the synaptic weights (memristance) that are collected in the columns. In particular, a neuron in the input layer injects a spike current s_i to the ith row. The spike current s_i is divided between the memristors of ith row. The lowest memristance g_{ij} in the ith row passes the largest spike current to the jth column. When the average of collected spike currents in a column is greater than others, the corresponding neuron generates spikes and becomes the winner.

C. Design Level III: COM Circuit

At this level, the hardware of COM is designed using the WTA modules and memristor crossbar arrays. COM is an assembly of WTAs with lateral excitatory synapses between the WTAs. The design of a WTA module has been introduced in Section III-B. In order to implement the lateral excitatory synapses, the memristor crossbar arrays are utilized. A memristor crossbar array is provided for each WTA module that conducts the spikes of the output neurons to the terminals T_{EX} of other WTA modules. Fig. 6 shows a COM architecture with three WTAs. An excitatory synapses bus is used to connect the outputs of ith WTA to inputs of jth WTA ($i \neq j$).

D. Training

In order to train a neuromorphic system, two main methods are available; *in situ* and *ex situ*. The *in situ* method utilizes extra circuitry on the chip to calculate the synaptic weights and program synaptic circuits. For instance, adjusting a memristor weight through the STDP algorithm requires a mechanism to send spikes of the postsynaptic neurons in direction of the presynaptic neurons [11]. In addition, the shape of the spike plays the main role to implement STDP for memristor-based synapses [48]. On the other hand, implementation of the *ex situ* method is independent of a training algorithm. The synaptic weights are calculated by a software out of the chip using a training algorithm. Then, the final weights are programmed to the synaptic circuits. The *ex situ* method requires an extra circuit to write/read a weight value to/from a synaptic circuit.

In our approach, the *ex situ* method is used to store a message in COM. At the pattern storage step, the STDP algorithm is performed in software to calculate the feedforward synaptic circuit between the input layer and output layers of a WTA. Also, the synaptic weights of the lateral excitatory synapses are calculated through the pattern association step. Then, the final weights are used to set the weights of the memristor crossbar arrays. Two steps are applied to set the memristor weights to the desired values, resetting and writing steps.

At the resetting step, all of the memristors are reset to the HRS that is illustrated in Fig. 7(a). To perform this process, a resetting voltage V_r is applied to all columns, while the rows are grounded. The voltage across all memristors becomes $-V_r$, which is less than the negative threshold voltage of a memristor (v_{thn}).

At the writing step, write pulses are applied to the rows in which the duration of the pulses is proportional to the desired weight vectors. Fig. 7(b) displays the process of writing a weight vector to a column of the memristor crossbar array. In order to change the memristance value, the amplitude of the applied pulse should be higher than the memristor threshold voltage. In order to protect other memristors from changing their memristors, a protective half-voltage is applied

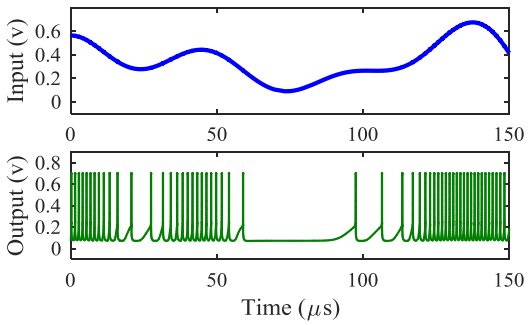

Fig. 8. Spiking neuron response to a random input.

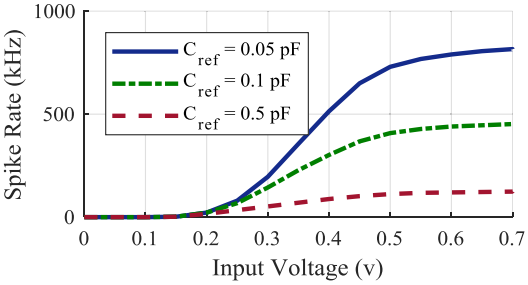

Fig. 9. Spike rate in response to different input voltages.

to the other columns [49]–[51]. The stochastic behavior of the memristor causes the important challenges for analog memristor programming [30], [52]. However, the memristors are programmed into HRS or LRS due to the bimodal values of the final synaptic weights.

IV. Results and Discussion

In this section, the simulation results and discussion are provided. The simulation results are reported for all design levels. First, the spiking neuron circuit is evaluated to approve its operation and to estimate its power consumption. The proposed neuron is also compared with the related works. Then, the WTA training and operation steps have been considered in our simulation. Finally, the simulation result of a COM is presented. In our applied simulations, the memristor model in [53] has been used which is compatible with the well-known models such as the fabricated memristor introduced by the University of Michigan [19].

A. Spiking Neuron

Spiking neuron receives the input voltage through the terminals of V_{FF}/V_{EX} and generates spike train in which the spike rate is proportional to the input amplitude. Fig. 8 illustrates the spikes train when a random voltage is applied to the input of the neuron. Higher amplitude generates higher spike rate. Fig. 9 shows the spike rate versus the applied voltage. When the input voltage V_{FF}/V_{EX} is less than the threshold voltage of $M1/M2$, the current I_{in} is negligible and spike rate becomes zero. In addition, refractory period time limits the maximum spike rate that is related to the capacitance of the C_{ref}. The simulation result demonstrated in Fig. 9 provides the effect of C_{ref} on maximum spike rate.

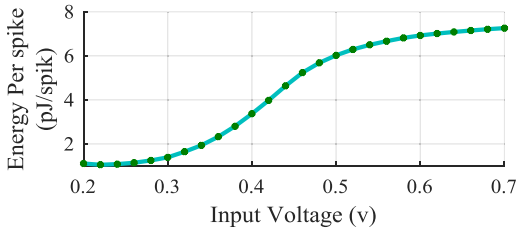

Fig. 10. Energy consumption per spike for the proposed neuron versus the applied voltage.

The static power consumption of the proposed neuron is about 182 pW which is a consequence of leakage of transistors. The energy consumption per spike is determined by

$$E = \frac{\int_{t1}^{t2} i(t).\mathrm{VDD}.dt}{N_{\mathrm{spk}}} \tag{6}$$

where $i(t)$ and V_{DD} are current and voltage of the power supply. The parameter of N_{spk} is the number of generated spikes during the period from t_1 and t_2.

In order to estimate the energy per spike, the simulation is performed for input voltage ranging from 0.2 to 0.7 V. In this regard, the input voltages are applied to the input of the circuit for 1 ms and the current of the supply voltage and the number of generated spikes are evaluated. Then, energy consumption per spike is calculated by (6). Fig. 10 indicates energy per spikes for different input voltages. The proposed neuron consumes energy per spike less than 7.3 pJ/spike, and the average is about 4.3 pJ/spike.

Table II shows the comparison of the proposed neuron with the related works regarding the important features such as energy consumption, circuit complexity, the capability to implement the refractory period mechanism, and lateral inhibitory interface. The energy consumption of the proposed neuron is comparable with the related works. The circuit in [10] consumes the least power. However, it is optimized to drive an active synaptic circuit where the synaptic circuit has no sink current. In addition, an output buffer should be used to drive an external equipment, where the buffer is not a part of the neuron [10]. The proposed neuron consumes more power; however, it is capable to drive a resistive load (memristor crossbar array).

The complexity is another aspect to compare the neuron circuits. The circuit complexity is evaluated based on the number of components and voltage sources. The proposed neuron includes 14 transistors and two capacitors, and it uses two different voltages. One of the voltages is used to supply the circuit and another to adjust the leakage current. Although the scheme in [54] includes 14 transistors and two capacitors, the circuit needs five voltage sources to supply the circuit and control the process.

Capability to implement the refractory period mechanism is another criterion to compare the works. Refractory period allows controlling the maximum spike rate that is implemented in the proposed neuron [9], [54] as well. Furthermore, the lateral inhibitory interface is necessary to implement a WTA module. The scheme in [55] employs digital components to

TABLE II
COMPARISON OF THE RELATED NEURON CIRCUITS

Related works	Energy Per Spike (pJ/spike)	Circuit complexity	Refractory period mechanism	lateral inhibitory interface
Indiveri, et al. [9]	900	#Transistor >22, #Capacitor=1, #Voltage source=2	Yes	No
Wijeksoon, et al. [54]	8.5 - 9	#Transistor =14, # Capacitor=2, #Voltage source=5	Yes	No
Jose, et al. [10]	0.4	#Transistor =16, # Capacitor=2, #Voltage source=4	No	No
Wu, et al. [55]	9.3	#Transistor >35, # Capacitor=3, # resistor = 3,	No	Yes (Digital interface)
This work	4.3	#Transistor =14, # Capacitor=2, #Voltage source=2	Yes	Yes (Analog interface)

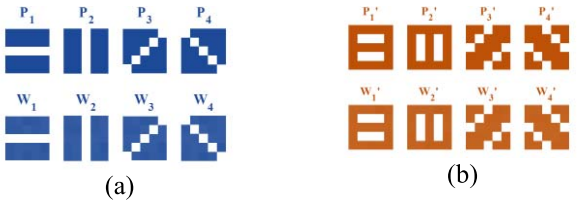

Fig. 11. Two sets of patterns (top) and corresponding synaptic weights (bottom). Pattern sets (a) and (b) are applied to the WTA_1 and WTA_2, respectively.

implement the lateral inhibitory such as flip-flop, OR gates, and tristate buffers. However, in the proposed neuron, a simple scheme is used to implement the lateral inhibitory interface.

B. Spiking WTA

The proposed neuron circuit is located in the input and output layers of the WTA. At the input layer, the neuron encodes the input pattern to spike trains. Consequently, spike trains are conducted to the output neurons through memristor crossbar array. The output neurons are connected to each other through the lateral inhibitory connections. The output neurons compete to generate spikes. The winner neuron represents the applied pattern.

Fig. 11 (top) illustrates two sets of patterns which are used to train two separate WTAs (WTA_1 and WTA_2). Each pattern is a 5×5 matrix, and the related WTA comprises $l = 25$ encoding neurons, $n = 4$ output neurons, and $k = 100$ synapses. Fig. 11 (bottom) provides the synaptic weights which are calculated through the STDP algorithm (pattern storage). The synaptic weights are similar to the corresponding pattern matrix. The synaptic weights are applied to the memristor crossbar array. The generated weights from Fig. 11(a) and (b) are applied to the memristor crossbar array of WTA_1 and WTA_2, respectively. Then, each pattern is applied to the corresponding WTA for 100 μs. Fig. 12(a) shows the responses of the neurons. A neuron is activated when the corresponding pattern is applied to the WTA. The pattern P_i is corresponding to the neuron n_i. Consequently, the other neurons are inhibited to generate spikes.

C. Message Storage and Retrieval in a COM

In Section IV-B, there are no lateral excitatory synapses between the WTAs. In order to implement a COM with two WTAs (WTA_1 and WTA_2), the weights of the lateral excitatory synapses are adjusted on the basis of messages. Four messages are used to store in the COM: $m_1 = \{p_1, p_1'\}, m_2 = \{p_2, p_2'\}, m_3 = \{p_3, p_3'\}$, and $m_4 = \{p_4, p_4'\}$. The synapses are then adjusted using (2) (pattern association). The patterns of Fig.11 (a) and (b) are applied to WTA_1 and WTA_2, respectively. Fig. 12(b) demonstrates the response of the neurons where the left figure and right figure are corresponding to WTA_1 and WTA_2, respectively. The neurons are activated properly. There are differences between Fig. 12(a) and (b) relating to the number of generated spikes. The number of spikes in Fig. 12(b) has been increased that is the effect of excitatory synapses. When a neuron n_i in WTA_1generates a spike, the excitatory synapses conduct it to the neuron n_i in WTA_2 and vice versa. Hence, the neuron n_i is stimulated through both input layer and lateral excitatory synapses. Consequently, the spike rate is increased. The main effect of the excitatory synapses is robust message retrieval. In order to show the robustness to the partially erased messages, the patterns of p_1, p_2, p_3, and p_4 in messages are erased by setting the values to zero. Then, the messages are applied to the COM. Consequently, the input of the WTA_1 is set to zero and the patterns of p_1', p_2', p_3', and p_4' are applied to the WTA_2. Fig. 12(c) shows the response of the neurons when messages are partially erased. Although the input of WTA_1 is zero, its neurons are activated through the excitatory synapses properly.

1) Robustness: In order to evaluate the robustness of the proposed architecture, random data sets have been used. In this regard, the number of $P = N \times n$ patterns with a length of l are randomly generated. Then, patterns are divided into N categories including n patterns. After that, M messages are produced by selecting a pattern of each category randomly. Noisy messages are created from the original messages by converting the elements of the original patterns from 1 to 0 and vice versa. The noise percentages of 10%, 15%, 20% and 30% are applied to the patterns. Furthermore, partially erased messages are also made from the original patterns. For each message, the partially erased messages are generated by erasing 20% and 40% of patterns in each message. Erasing a pattern is performed by setting the values of the pattern elements to zero. In addition, the imperfections of memristor crossbar array are considered by adding process variation effect and SAFs on the memristor crossbar. In order to apply the effect of the process variation, the internal state x of the memristor is disturbed according to the Gaussian distribution with the standard deviation of 0.1 and mean of 0. Moreover, SAFs (stuck-at-zero and stuck-at-one) are randomly inserted into the memristor crossbar array. The impact of 5% and 10% of SAFs are considered for all of the memristor crossbar arrays.

Fig. 13 shows the simulation results for two architectures of COM. The simulation result for COM with $N = 4, n = 4, l = 30$, and 10 messages is illustrated

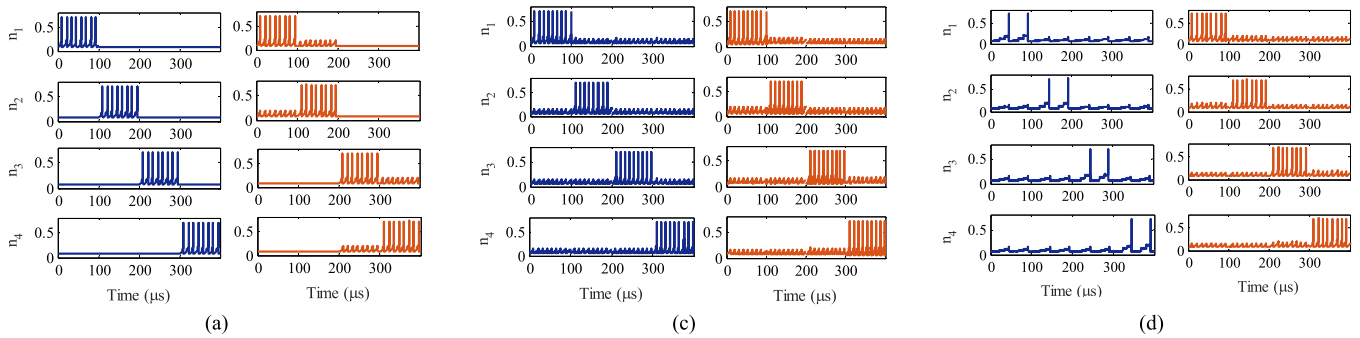

Fig. 12. Response of neurons of WTA_1 (WTA_2) to the applied patterns from Fig. 11(a) [Fig. 11(b)]. (a) Two separate WTA modules. (b) COM with two WTA modules, when original messages are applied to the COM. (c) COM with two WTA modules, when partially erased messages are applied to the COM.

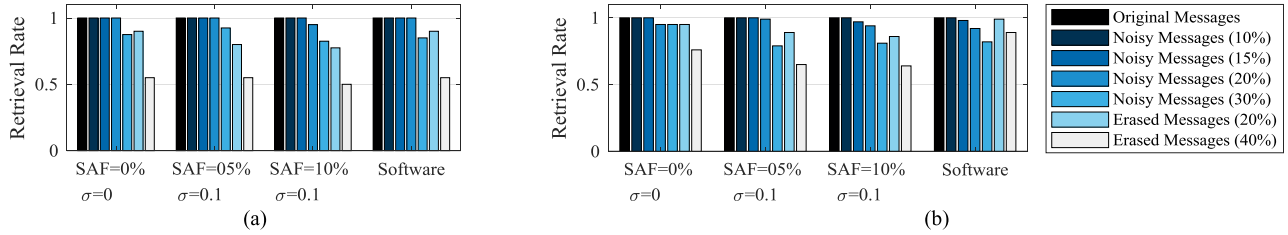

Fig. 13. Retrieval rate for different architectures of a COM in the presence of message noise, message erasure, SAFs, and process variation. (a) COM with $N = 4$, $n = 4$, and $l = 30$. (b) COM architecture with $N = 5$, $n = 8$, and $l = 40$.

in Fig. 13(a). In this simulation, 16 patterns are generated randomly. Then, the patterns are divided into four sets. Each set is used to train a WTA (pattern storage), and the feedforward synaptic circuit is adjusted. Ten messages are generated randomly and applied to adjust the excitatory synaptic circuits (pattern association). The noisy (10%, 15%, 20%, and 30%) and erased messages (20% and 40%) are generated using the original messages. In addition, different impacts of the SAFs (0%, 5%, and 10%) and process variation ($\sigma = 0, \sigma = 0.1$) have been applied to all of the memristor crossbar arrays. Finally, all of the messages (original, noisy, and erased messages) have been used to evaluate the retrieval rate. The retrieval rate is calculated by N_a/N, where N_a is the number of correct activated neurons and N is the number of the activated neurons (i.e., the number of WTAs in a COM). All of the mentioned steps are performed for a COM architecture with $N = 5, n = 8, l = 40$, and 20 messages accordingly. The simulation results are shown in Fig. 13(b).

The results show the robustness of the proposed architecture for noisy patterns, erased patterns, SAFs, and process variation. The noisy messages of 10% and 15% are retrieved completely even in the presence of SAFs (5% and 10%) and process variation ($\sigma = 0.1$). However, the retrieval of the noisy messages of 20% and 30% and partially erased messages are degraded. The average of the message retrieval rate in hardware simulation is 0.9. In addition, the average of the message retrieval rate in the corresponding software-based simulation is 0.92.

2) Capacity: Large capacity is another promising feature of a COM. The upper bound of capacity is calculated by [5]

$$C_{\text{Max}} = \left(\rho \frac{2^{l+1}}{(l+1)} + 1\right)^{N}. \tag{7}$$

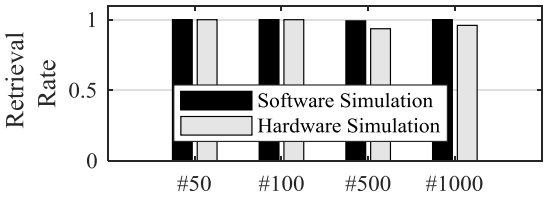

Fig. 14. Retrieval rate for different number of messages.

where ρ is a small real value (for instance, 0.01), which is related to the accepted error in the message retrieval.

A COM architecture with $N = 5, n = 4$, and $l = 20$ is employed to prove the capability of the proposed architecture for storing and retrieving a large number of messages. The number of $P = n \times N$ patterns and M messages is generated similar to the previous section ("Robustness"). The noisy messages (15%) are generated using 5% of the original messages. Then, the original patterns and messages are used to train COM. The noisy messages are applied to the tainted COM in the presence of SAFs (5%) and the process variations ($\sigma = 0.1$). Fig. 14 shows the retrieval rate for a different number of messages M (50, 100, 500, and 1000). The software-based simulation retrieval rate is almost 1 and the hardware simulation is about 0.97 due to the memristor imperfections.

V. Conclusion

In this paper, a new hardware architecture for a COM based on spiking neuron and memristor crossbar arrays is proposed. A COM contains WTAs with excitatory synapses. In order to implement a COM, three design levels have been employed. At design level I, a low power spiking neuron circuit is

proposed. At design level II, the proposed neuron and memristor crossbar arrays are used to implement a WTA module. At design level III, the WTA modules and the memristor crossbar arrays have been used to implement the COM architecture. The simulation results prove the proper operation of the proposed neuron, WTA module, and COM. The robustness and capacity of the COM hardware are evaluated and compared with the software-based simulation. Random messages are applied to evaluate the message retrieval of the COM. The average of message retrieval rate of the hardware and software-based simulation is about 0.94 and 0.96, respectively.

REFERENCES

[1] S. B. Furber, "Brain-inspired computing," *IET Comput. Digit. Techn.*, vol. 10, no. 6, pp. 299–305, Nov. 2016.

[2] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.

[3] B. Kosko, "Adaptive bidirectional associative memories," *Appl. Opt.*, vol. 26, no. 23, pp. 4947–4960, Dec. 1987.

[4] V. Gripon and C. Berrou, "Sparse neural networks with large learning diversity," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1087–1096, Jul. 2011.

[5] J. Shamsi, K. Mohammadi, and S. B. Shokouhi, "Columnar-organized memory (COM): Brain-inspired associative memory with large capacity and robust retrieval," *Biol. Inspired Cognit. Archit.*, vol. 20, pp. 39–46, Apr. 2017.

[6] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.

[7] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA, USA: Addison-Wesley, 1989.

[8] A. van Schaik, "Building blocks for electronic spiking neural networks," *Neural Netw.*, vol. 14, nos. 6–7, pp. 617–628, Jul. 2001.

[9] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 211–221, Jan. 2006.

[10] J. M. Cruz-Albrecht, M. W. Yung, and N. Srinivasa, "Energy-efficient neuron, synapse and STDP integrated circuits," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 3, pp. 246–256, Jun. 2012.

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

[12] J. Shamsi, K. Mohammadi, and S. B. Shokouhi, "A low power circuit of a leaky integrate and fire neuron with global reset," in *Proc. Iranian Conf. Elect. Eng. (ICEE)*, 2017, pp. 366–369.

[13] J. Shamsi, A. Amirsoleimani, S. Mirzakuchaki, and M. Ahmadi, "Modular neuron comprises of memristor-based synapse," *Neural Comput. Appl.*, vol. 28, no. 1, pp. 1–11, Jan. 2017.

[14] L. D. Jackel, H. P. Graf, and R. E. Howard, "Electronic neural network chips," *Appl. Opt.*, vol. 26, no. 23, p. 5077, Dec. 1987.

[15] S. Mitra, S. Fusi, and G. Indiveri, "Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 1, pp. 32–42, Feb. 2009.

[16] M. R. Azghadi, N. Iannella, S. F. Al-Sarawi, G. Indiveri, and D. Abbott, "Spike-based synaptic plasticity in silicon: Design, implementation, application, and challenges," *Proc. IEEE*, vol. 102, no. 5, pp. 717–737, May 2014.

[17] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Front. Neurosci.*, vol. 5, p. 73, May 2011.

[18] S. Ramakrishnan, P. E. Hasler, and C. Gordon, "Floating gate synapses with spike-time-dependent plasticity," *IEEE Trans. Biomed. Circuits Syst.*, vol. 5, no. 3, pp. 244–252, Jun. 2011.

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

[20] S. Choi, P. Sheridan, and W. D. Lu, "Data clustering using memristor networks," *Sci. Rep.*, vol. 5, Jan. 2015, Art. no. 10492.

[21] C. Yakopcic, R. Hasan, T. M. Taha, M. McLean, and D. Palmer, "Memristor-based neuron circuit and method for applying learning algorithm in SPICE?" *Electron. Lett.*, vol. 50, no. 7, pp. 492–494, Mar. 2014.

[22] M. Imani, A. Rahimi, D. Kong, T. Rosing, and J. M. Rabaey, "Exploring hyperdimensional associative memory," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2017, pp. 445–456.

[23] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1864–1878, Oct. 2014.

[24] X. Guo *et al.*, "Modeling and experimental demonstration of a hopfield network analog-to-digital converter with hybrid CMOS/memristor circuits," *Front. Neurosci.*, vol. 9, p. 488, Dec. 2015.

[25] B. Nessler, M. Pfeiffer, L. Buesing, and W. Maass, "Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity," *PLoS Comput. Biol.*, vol. 9, no. 4, p. e1003037, Apr. 2013.

[26] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers Comput. Neurosci.*, vol. 9, p. 99, Aug. 2015.

[27] A. Tavanaei and A. S. Maida, "A minimal spiking neural network to rapidly train and classify handwritten digits in binary and 10-digit tasks," *Int. J. Adv. Res. Artif. Intell.*, vol. 4, no. 7, pp. 1–8, 2015.

[28] M. Al-Shedivat, R. Naous, G. Cauwenberghs, and K. N. Salama, "Memristors empower spiking neurons with stochasticity," *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 5, no. 2, pp. 242–253, Jun. 2015.

[29] T. Iakymchuk, A. Rosado-Muñoz, J. F. Guerrero-Martínez, M. Bataller-Mompeán, and J. V. Francés-Víllora, "Simplified spiking neural network architecture and STDP learning algorithm applied to image classification," *EURASIP J. Image Video Process.*, vol. 2015, no. 1, p. 4, Dec. 2015.

[30] J. Bill and R. Legenstein, "A compound memristive synapse model for statistical learning through STDP in spiking neural networks," *Front. Neurosci.*, vol. 8, p. 412, Dec. 2014.

[31] T. Masquelier, R. Guyonneau, and S. J. Thorpe, "Competitive STDP-based spike pattern learning," *Neural Comput.*, vol. 21, no. 5, pp. 1259–1276, May 2009.

[32] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.

[33] Y. Dan and M.-M. Poo, "Spike timing-dependent plasticity: From synapse to perception," *Physiol. Rev.*, vol. 86, no. 3, pp. 1033–1048, Jul. 2006.

[34] Y. A. Vlasov and S. J. McNab, "Coupling into the slow light mode in slab-type photonic crystal waveguides," *Opt. Lett.*, vol. 31, no. 1, p. 50, Jan. 2006.

[35] S. Gaba, P. Sheridan, J. Zhou, S. Choi, and W. Lu, "Stochastic memristive devices for computing and neuromorphic applications," *Nanoscale*, vol. 5, no. 13, pp. 5872–5878, Jul. 2013.

[36] J. A. Starzyk and Basawaraj, "Memristor crossbar architecture for synchronous neural networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 8, pp. 2390–2401, Aug. 2014.

[37] J. Rajendran, R. Karri, and G. S. Rose, "Improving tolerance to variations in memristor-based applications using parallel memristors," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 733–746, Mar. 2015.

[38] B. Liu, H. Li, Y. Chen, X. Li, Q. Wu, and T. Huang, "Vortex: Variation-aware training for memristor X-bar," in *Proc. 52nd Annu. Design Autom. Conf. (DAC)*, 2015, pp. 1–6.

[39] D. Chabi, Z. Wang, C. Bennett, J.-O. Klein, and W. Zhao, "Ultrahigh density memristor neural crossbar for on-chip supervised learning," *IEEE Trans. Nanotechnol.*, vol. 14, no. 6, pp. 954–962, Nov. 2015.

[40] Z. Fang, H. Y. Yu, X. Li, N. Singh, G. Q. Lo, and D. L. Kwong, "$HfO_x/TiO_x/HfO_x/TiO_x$ multilayer-based forming-free RRAM devices with excellent uniformity," *IEEE Electron Device Lett.*, vol. 32, no. 4, pp. 566–568, Apr. 2011.

[41] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H.-S. P. Wong, "A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation," *Adv. Mater.*, vol. 25, no. 12, pp. 1774–1779, Mar. 2013.

[42] E. Sugawara and H. Nikaido, "Properties of AdeABC and AdeIJK efflux systems of *Acinetobacter baumannii* compared with those of the AcrAB-TolC system of *Escherichia coli*," *Antimicrobial Agents Chemotherapy*, vol. 58, no. 12, pp. 7250–7257, Dec. 2014.

[43] L. Chen *et al.*, "Accelerator-friendly neural-network training: Learning variations and defects in RRAM crossbar," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 19–24.

[44] C.-Y. Chen *et al.*, "RRAM defect modeling and failure analysis based on march test and a novel squeeze-search scheme," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 180–190, Jan. 2015.
[45] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectron. J.*, vol. 44, no. 2, pp. 176–183, 2013.
[46] B. Liu *et al.*, "Reduction and IR-drop compensations techniques for reliable neuromorphic computing systems," in *IEEE/ACM Int. Conf. Comput.-Aided Design, Dig. Tech. Papers (ICCAD)*, Jan. 2015, pp. 63–70.
[47] J. Liang and H.-S. P. Wong, "Cross-point memory array without cell selectors-device characteristics and data storage pattern dependencies," *IEEE Trans. Electron Devices*, vol. 57, no. 10, pp. 2531–2538, Oct. 2010.
[48] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, "STDP and STDP variations with memristors for spiking neuromorphic learning systems," *Frontires Neurosci.*, vol. 7, p. 2, Feb. 2013.
[49] L. Vinet and A. Zhedanov, "A 'missing' family of classical orthogonal polynomials," *J. Phys. A, Math. Theor.*, vol. 44, no. 8, p. 85201, Feb. 2011.
[50] F. Alibart, E. Zamanidoost, and D. B. Strukov, "Pattern classification by memristive crossbar circuits using *ex situ* and *in situ* training," *Nat. Commun.*, vol. 4, p. 2072, Jun. 2013.
[51] P. M. Sheridan, C. Du, and W. D. Lu, "Feature extraction using memristor networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2327–2336, Nov. 2016.
[52] S. N. Mozaffari, S. Tragoudas, and T. Haniotakis, "More efficient testing of metal-oxide memristor–based memory," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 36, no. 6, pp. 1018–1029, Jun. 2017.
[53] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Generalized memristive device SPICE model and its application in circuit design," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 32, no. 8, pp. 1201–1214, Aug. 2013.
[54] J. H. B. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and bursting behaviour," *Neural Netw.*, vol. 21, nos. 2–3, pp. 524–534, Mar. 2008.
[55] X. Wu, V. Saxena, K. Zhu, and S. Balagopal, "A CMOS spiking neuron for brain-inspired neural networks with resistive synapses and *in situ* learning," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 11, pp. 1088–1092, May 2015.

Jafar Shamsi received the M.Sc. degree in electrical engineering from the Iran University of Science and Technology, Tehran, Iran, in 2013, where he is currently working toward the Ph.D. degree in electronic engineering.

His current research interests include brain-inspired architectures and neuromorphic and memristor-based circuit designs.

Karim Mohammadi received the B.S. degree in electrical engineering from the Iran University of Science and Technology, Tehran, Iran, in 1972, the M.Sc. degree in electrical engineering from Wayne State University, Detroit, MI, USA, in 1978, and the Ph.D. degree in electrical engineering from Oakland University, Rochester, MI, USA, in 1981.

He is currently a Professor of Electrical Engineering at the Iran University of Science and Technology. His current research interests include reconfigurable systems, fault-tolerant systems, reliable computing, digital systems, and microprocessors.

Shahriar B. Shokouhi (M'10) received the B.Sc. and M.Sc. degrees in electrical and electronic engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 1986 and 1989, respectively, and the Ph.D. degree in electronic and electrical engineering from the University of Bath, Bath, U.K., in 1999.

He was a Research Associate at the Department of Electrical and Computer Engineering, Western University, London, ON, Canada, from 2012 to 2014. He is currently a Faculty Member of the Electrical Engineering Department at IUST. His current research interests include embedded systems, trusted hardware designs, machine vision, and intelligent system designs.

Dr. Shokouhi is a member of the Circuit and Systems Society.