Arabian Journal for Science and Engineering (2023) 48:6175–6183
https://doi.org/10.1007/s13369-022-07290-7

RESEARCH ARTICLE-ELECTRICAL ENGINEERING

The EGM Model and the Winner-Takes-All (WTA) Mechanism for a Memristor-Based Neural Network

Mouna Elhamdaoui[1] · Faten Ouaja Rziga[1] · Khaoula Mbarek[1] · Kamel Besbes[1,2]

Received: 16 February 2022 / Accepted: 8 September 2022 / Published online: 3 October 2022

Abstract
Due to the continuous growth of hardware neuromorphic systems, the need for high-speed, low-power, and energy-efficient computer architectures is increasing. Memristors-based neural networks are a promising solution for low-power neuromorphic systems. Spiking neural networks (SNNs) have been considered the optimal hardware implementation of these systems. Previous studies of SNNs rely on complex circuit to implement in situ bio-plausible STDP learning using memristors, which is computationally challenging. In this paper, we propose an SNN that performs both in situ learning and inference using a new efficient programming technique. Our interest lies in applying the winner-takes-all (WTA) mechanism in the SNN architecture used with recurrently connected neurons, allowing real-time processing of patterns. We provide a programming circuit that enables better weight modulation with less power consumption and less space occupation, using a generalized enhanced memristor model (EGM). The proposed programming circuit is connected to leaky integrate-and-fire (LIF) neurons included in a crossbar architecture to perform recognition task. The simulation results not only prove the correctness of the design, but also offer an efficient implementation in terms of area, energy, accuracy, as well as the ability to classify 40,000 images per second.

Keywords Memristor · EGM model · Neural network · Synapse · SNN · WTA · Patterns recognition

1 Introduction

Over the last few decades, spiking neural networks (SNNs) have been developed as the third generation of artificial neural networks (ANNs) [1]. It is a promising neural network that is inspired by the functionality of the human brain, enabling spike-based temporal processing and offer low power consumption [2, 3].

SNN is developed based on spiking neurons and plastic synapses, where the inputs and outputs of the neurons are encoded as time series of spikes. The leaky integrate-and-fire (LIF) neuron model, out of all the neuron models, provides the best balance of accuracy and ease of hardware implementation, and it also closely matches biological neurons [4, 5].

✉ Faten Ouaja Rziga
faten_ouaja@yahoo.fr

[1] Microelectronics and Instrumentation Lab, University of Monastir, Monastir, Tunisia

[2] Centre for Research on Microelectronics & Nanotechnology Sousse Technopole, Sousse, Tunisia

The other crucial block of the SNN is the synaptic circuit that defines the strength of the connection between neurons and stores synaptic weight.

In traditional SNN, researchers have employed VLSI technology consisting mainly of complementary metal–oxide–semiconductor (CMOS) devices to implement the synaptic devices. However, the implementation of CMOS technology for such circuits requires a large number of transistors, which requires a large area and high-power consumption [6, 7].

Memristor has the potential to become the most suitable candidate for emulating a synapse with smaller on-chip area and low power dissipation [8]. It provides a tunable and non-volatile storage of synaptic weights [9–11].

Currently, there is a growing interest in integrating memristive synapses to build multiple SNN architectures [12–15]. The learning process used to adjust and update the synaptic weights is an important aspect of the neural network. The backpropagation training algorithm used in the literature [6, 7] is well established but it requires significant hardware resources and it is difficult to fully implement in analog circuits.

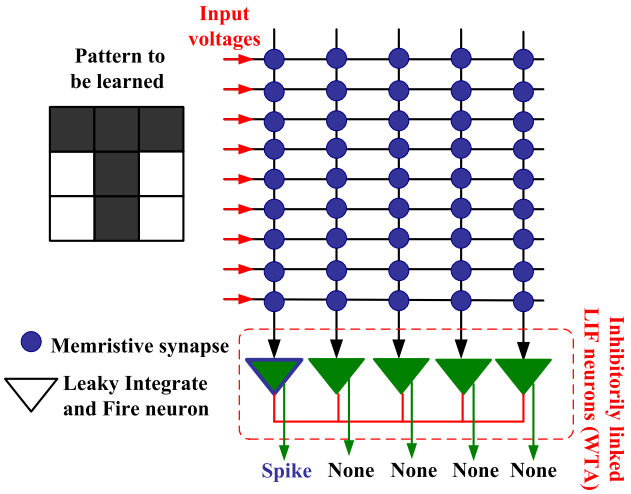

Fig. 1 Proposed spiking neural network comprising input layer connected to output layer, with five LIF neurons, by a memristive crossbar

Several published studies have demonstrated the training of memristive neural networks using the ex-situ method [16] in which, weight calculation is performed by an external circuit or computer platforms, which increases the circuit complexity and power consumption.

A part of this learning method, the in situ method has been commonly performed. Most previous implementations of this method have been limited on spike-timing-dependent plasticity (STDP) learning [12]. However, they may require extra training circuitry attached to the synapses, or different waveforms for pre- and post-synaptic spikes [17].

During inference in SNN, the input will be converted to a read voltage and applied to the pre-trained memristor weight. Thus, the more efficient way to realize a real challenging memristive neural network is to do both training and inference in hardware.

In this paper, we propose a memristor-based SNN that rely on a WTA mechanism [18] to perform an unsupervised on chip learning for real-time recognition task.

This SNN-based WTA architecture consists of a 9×5 memristive crossbar with nine inputs, programming synaptic circuit and five competitive LIF output neurons, as shown in Fig. 1. The output LIF neurons will compete with each other at the output layer in order to identify the winner. Thereby, the winner indicates which of the learned patterns was displayed on the network input.

In order to successfully adjust the memristor weight, we propose a simple and efficient programming circuit using an enhanced and generalized memristor model (EGM) which is capable of performing both learning and inference in hardware circuit. Therefore, this new programming circuit enables an effective modulation of the memristor conductance without any processing circuitry, considering not only the complexity and weight adjustment ability of the synaptic circuit but also the power and space consumption.

Hence, using our proposed architecture, we successfully validated a real-time recognition task of 3×3 pixel images. A series of simulations are performed to verify the design. We prove that this network is tolerable to noisy patterns and offer an efficient implementation in terms of area, energy, accuracy, as well as the ability to classify 40,000 images per second.

This paper consists of the following sections: In Sect. 2, we present the mathematical description of the EGM model. We explore the I–V characteristic of the EGM with respect to the experimental data, as well as its conductance change in order to validate the potentiation and depression mechanisms. We also present the proposed programming technique implemented in SNN to simulate both learning and inference in situ. In Sect. 3, we introduce the LIF circuit applied in the output layer used to implement lateral inhibition in the WTA network. Section 4 demonstrates the proposed SNN architecture implemented using the WTA mechanism. This network is simulated and validated in a real-time images recognition application. Section 5 concludes the paper.

2 Memristor-Based Synapse

Memristor is a two-terminal storage element that can exhibit dynamic reconfiguration within the application of electrical stimuli, resulting in a modulation of the resistance known as the memory effect [19, 20].

The changed resistance state can be retained even after the electrical input is removed. These capabilities conduct analog switching, which is similar to biological synapses where the strength (or synaptic weight) can increase or decrease depending on the applied action potential [21]. In addition, several beneficial properties of memristors have been demonstrated, including a nanoscale footprint [22], long endurance and retention [23], fast switching speed in the nanosecond range [24], and low energy consumption [24]. These characteristics have made memristors promising candidates for artificial synapses.

Memristive synapses have possible implementation in the crossbar array structure to realize multiple neuromorphic tasks. The crossbar structure further allows physical mapping of the neural network in hardware and facilitates parallel processing of computationally expensive matrix operations directly in memory [25]. Moreover, the conductance of the memristor can be flexibly tuned by modulating the parameters of the applied voltage pulses, thus providing great potential to construct adaptive systems with online learning capability. These desirable properties make memristors particularly attractive as neuromorphic devices (i.e., synapses and neurons). The emergence of memristors and

Table 1 The implemented parameters used in the EGM model

Parameter	Value	Parameter	Value
a_1	0.05	A_n	6000
a_2	0.05	x_p	0.3
b	0.05	x_n	0.7
V_p	0.75 [V]	α_p	0.5
V_n	0.75 [V]	α_n	1
A_p	6000	x_0	0.1

their synaptic-like behavior opened the possibility to overcome the limitations of CMOS technologies. Memristors can be a few nanometers size and can be packed densely in a two-dimensional layer with nanometer-range pitch, potentially offering higher neuron and synaptic density. With a fabrication process much cheaper than CMOS [26].

2.1 EGM Simulation

The memristor model used throughout this paper is a voltage-controlled threshold model, named the Enhanced Generalized Memristor model (EGM) [27]. It is flexible, general and can be adapted to different memristive devices [10]. This model is developed based on a SPICE model proposed by Yakopcic et al. [28]. Then, it was modified and implemented in Verilog-A [27].

The EGM model is practically based on the two mathematical functions presented in Eqs. (1) and (2).

$$i(t) = \begin{cases} a_1 x(t) \sinh(bv(t)); & v(t) \geq 0 \\ a_2 x(t) \sinh(bv(t)); & v(t) < 0 \end{cases} \tag{1}$$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \eta g(v(t)) f(x(t)) \tag{2}$$

Equation (1) gives the relationship between the memristor voltage and the memristor current. Then, Eq. (2) gives the derivative of the model state variable. Where $g(V(t))$ and $f(x(t))$ are two functions used to implement the memristor threshold voltage parameters *Vp* and *Vn*, and limit the change of device state when the state variable x approaches the boundaries.

For further mathematical details, you can refer to our previous articles [10, 27].

We use the Spectre circuit in Virtuoso® Custom IC Design to simulate the EGM model, where Table 1 gives its simulation parameters.

Figure 2a displays the current I and voltage V curves of the EGM model matching the physical data of the crystalline GST-based memristor device [29]. We notice that the I–V characteristics obtained from the EGM model have a symmetrical waveform which agrees well with the experimental data of the GST memristor device [29].

This model is used to determine synaptic behavior, Fig. 2b demonstrates the device response for both potentiation and depression processes after application of a positive and negative input voltage. The amplitudes of the depressing and potentiating pulses are + 1 V and − 1 V, respectively, applied for 10 µs pulse width.

It is evident that the device response can be adjusted depending on the amplitude of stimulation pulses, indicating stable and uniform potentiation and depression beyond the polarity of the applied bias can be used to emulate the weight adjustment of a memristive synapse.

2.2 Programming Circuit

We designed a single memristor synapse (1 M) implemented in a programming circuit to perform the memristor conductance modulation in the network. This circuit able to simulate both learning and inference in situ, where each synapse stores a single synaptic weight W_{ij}. Several programming techniques are used to accurately modulate the memristor. These techniques are designed to increase or decrease the memristor conductance to a predefined value, using pulses of distinct amplitude and time width, for reading, writing, and resetting the device to an intermediate state.

These techniques suffer from the drawback of requiring external processing to fine-tune the memristor.

In this paper, we propose a simple and efficient circuit using a pulse programming approach, that does not suffer from the need for external processing. The block diagram of the programming circuit is given in Fig. 3a.

The circuit consists of a memristor implemented using the EGM model, four selectors implemented using NMOS transistors, and four controlled pulses $\mathrm{VS_{write}}$, $\mathrm{VS_{read}}$, V_{write} and V_{read}. Since the network implements both learning and inference in situ, the circuit must efficiently support these two processes. The learning process consists of modulating the weight of the synapse to the desired value as a function of the applied voltage. Whereas during inference process, the input will be converted to a read voltage. These processes are applied to program the 1 M synapse, as write and read phases. The two selectors $\mathrm{S1_{write}}$ and $\mathrm{S2_{write}}$ are responsible for the write phase and the selectors $\mathrm{S1_{read}}$ and $\mathrm{S2_{read}}$ are responsible for the read phase. $\mathrm{VS_{write}}$ and $\mathrm{VS_{read}}$ are two controlled voltages used to control the switching of the selectors. V_{write} is used to program the conductance of the memristor. During the write mode, $\mathrm{S1_{write}}$ and $\mathrm{S2_{write}}$ are activated and $\mathrm{S1_{read}}$ and $\mathrm{S2_{read}}$ are turned off. Therefore, the memristor is connected to V_{write} and a programming pulse with a proportional duration and amplitude to the desired weight vector were

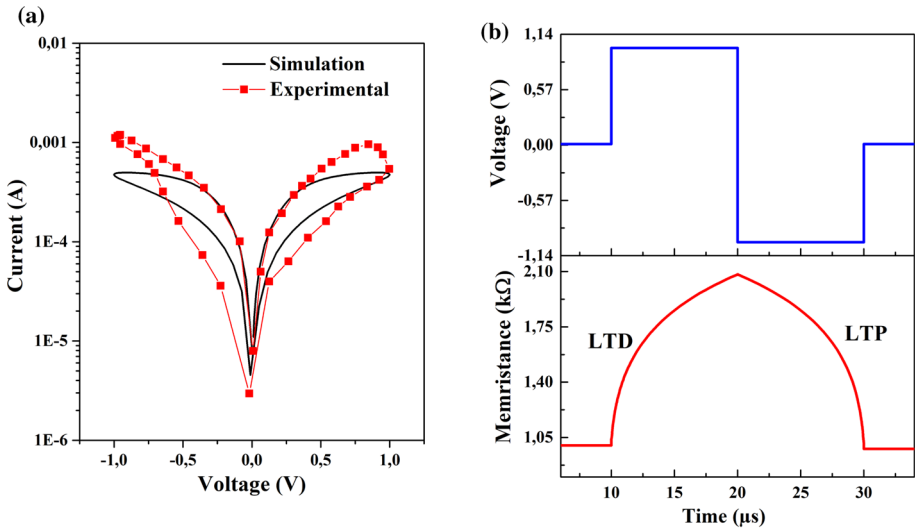

Fig. 2 EGM synapse simulation **a** I–V characteristic of the EGM model, **b** potentiation and depression under the influence of positive and negative pulses

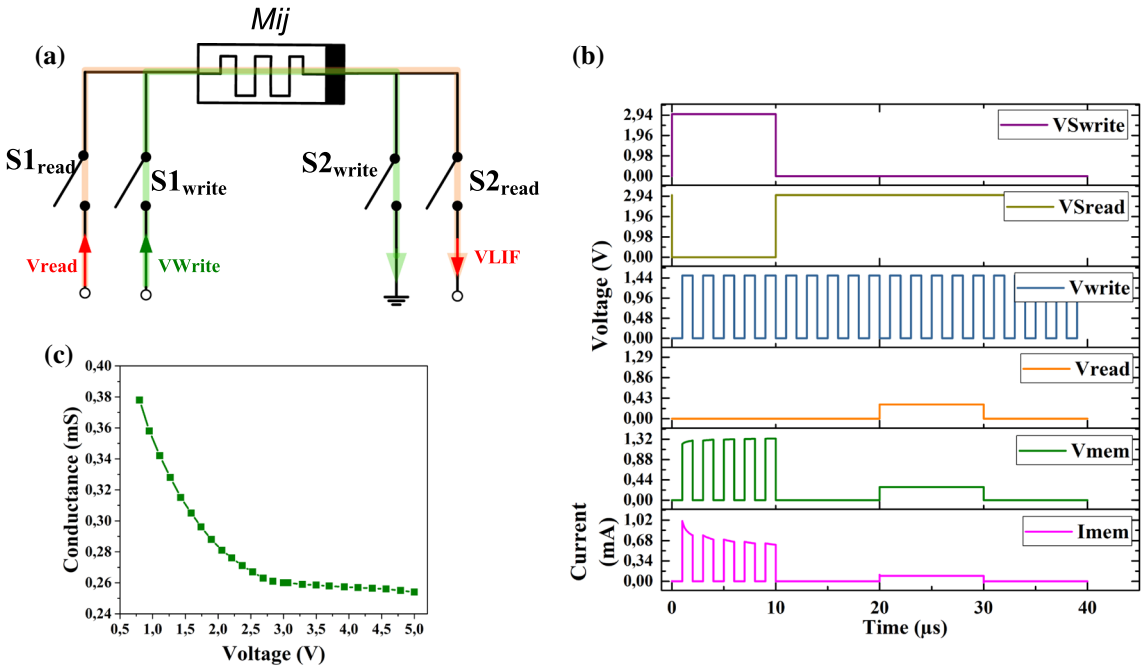

Fig. 3 **a** Functional block of the proposed programming circuit, **b** simulation results of the programming circuit, **c** programmed conductance as a function of the different applied voltage Amplitude V_{Write}

applied to the EGM terminals. In order to change the conductance value, the amplitude of the applied pulse must be greater than the threshold voltage of the memristor model which is equal to 0.75 V. Then, during read mode, $S1_{\text{read}}$ and $S2_{\text{read}}$ are turned on and $S1_{\text{write}}$ and $S2_{\text{write}}$ are turned off. As a result, the memristor is connected to V_{read}. This phase requires a read voltage of a magnitude lower than the threshold voltage of the memristor to ensure that the conductance state remains unchanged. The simulation results of the programming circuit are shown in Fig. 3b. By applying a write voltage of 1.5 V and a pulse width of 1 µs, and a read voltage of 0.3 V, we obtained a current value of 89.7 µA corresponding to a memristor conductance of 29.9 µS.

The operation of the proposed circuit is validated for multiple input voltages. We first change the amplitude of the writing voltage and then we read the value of the conductance for each applied voltage.

Figure 3c illustrates the conductance curve for different values of pulse amplitude applied to the memristor terminal ranging from 0.8 to 5 V, which gives a variation of conductance from 37.8 to 25.4 µS.

These results validate the ability of our proposed circuit to program the memristor conductance in different states. This

leads us to implement it in the WTA architecture to ensure unsupervised learning.

As mentioned earlier, the network is trained on black and white images of size 3×3 pixels, so each of the nine inputs of the WTA network is encoded with a voltage pulse according to the pixels in the images. Our methodology requires the use of two conductance values, where one value is assigned to white pixels and the other to black pixels.

3 Leaky Integrate-and-Fire Neuron (LIF)

The leaky integrate-and-fire (LIF) circuit used in this work was proposed in [30]. The LIF neuron model has been widely explored in the literature. The main reasons for using the LIF model in our architecture are as follows;

- The LIF circuit works efficiently in spiking and event-driven networks,
- It connects to the 1 M synapse at only one terminal
- The potential to reset its membrane potential by an external global reset.
- It has a low energy consumption (energy per spike is about 2 pJ/spike) and a fast response time (around 6 μs).

This LIF neuron serves also as an encoding unit, converting an input current into a spike train. The spike train's rate is proportional to the current input.

As shown in Fig. 4a, the LIF circuit has several sections, including a leaky integrator, a spike generator and a global reset. The leaky integrator contains a current mirror and a capacitor C_u for leakage and current integration. For the spike generation, a Schmitt trigger and a buffer are used. T_{11}–T_{12} are used for the reset process, and the capacitor C_{ref} for the refractory period. The input current I_{in} is injected into the leaky integrator through the current mirror block. Then, this current is integrated by C_u and it is leaked through T_3. When the membrane potential reaches the switching voltage of the Schmitt trigger, the output node o switches from high to low state. Consequently, T_{10} is turned on and the membrane potential is reset through T_4. As a result, a spike is generated at the neuron output. The switching voltage of the Schmitt trigger is firing threshold of the neuron that is determined by Eq. (3):

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

where $R_{np} = \sqrt{\frac{\beta_{T7}}{\beta_{T5}}}$, $R_{nn} = \sqrt{\frac{\beta_{T9}}{\beta_{T7}}}$, βT_i corresponds to the transconductance of transistors T_i, to simplify, the threshold of all memristors is considered equal to V_{th}.

In addition, the membrane potential can be reset via the lateral inhibitory terminal G_{rst}, which is used to implement lateral inhibition in the spiking WTA.

Figure 4 (b) shows the simulation results of the LIF neuron including the curves of the input current, the membrane potential (voltage across the capacitor C_u), and the output spike train of the LIF neuron. The Schmitt trigger switching threshold voltage was set to 644 mV.

The duration of the output spike is 10 ns and its amplitude is 0.9 V. We note that a higher spike rate is produced by a larger amplitude of the input current. These results prove that the LIF neuron is sustainable in our network.

4 Memristor-Based SNN System

4.1 Circuit Design

The proposed SNN is designed using a 9×5 crossbar array, with nine inputs and five outputs as described in Fig. 5. It performs the unsupervised learning with WTA functionality in situ to recognize five patterns T, X, V, C and L. This architecture aims to explore the relationship between the feature of an individual EGM model and the overall system performance in a neural network. Each input is encoded by voltage pulses on a pixel of 3×3 images containing one of the five patterns to be classified.

The conductance of each memristor model is modulated (learned) according to the learning patterns, which is performed through a programming circuit as shown in Fig. 3. This circuit consists of one memristor and four selectors that enable the learning and inference processes. The memristive crossbar is capable of performing matrix multiplication according to Ohm's and Kirchhoff's current laws.

LIF neurons are employed on the output layer, with adaptive thresholds for stimulation and global reset. They are laterally connected within an inhibitory WTA network. These neurons emit spikes based on the particular conductance of the memristor models and the applied voltage pulses represent the five input patterns. They integrate the input signals that pass through the memristors along each row and fire when a certain threshold voltage is reached. When an output neuron generates a spike, the membrane potential of the other neurons is reset to zero by the global reset.

This enables unsupervised learning since each output neuron creates its appropriate receptive field during the learning process. Thereafter, in the recognition phase, each of the output neurons generates a spike based on the previously trained pattern for a variable input.

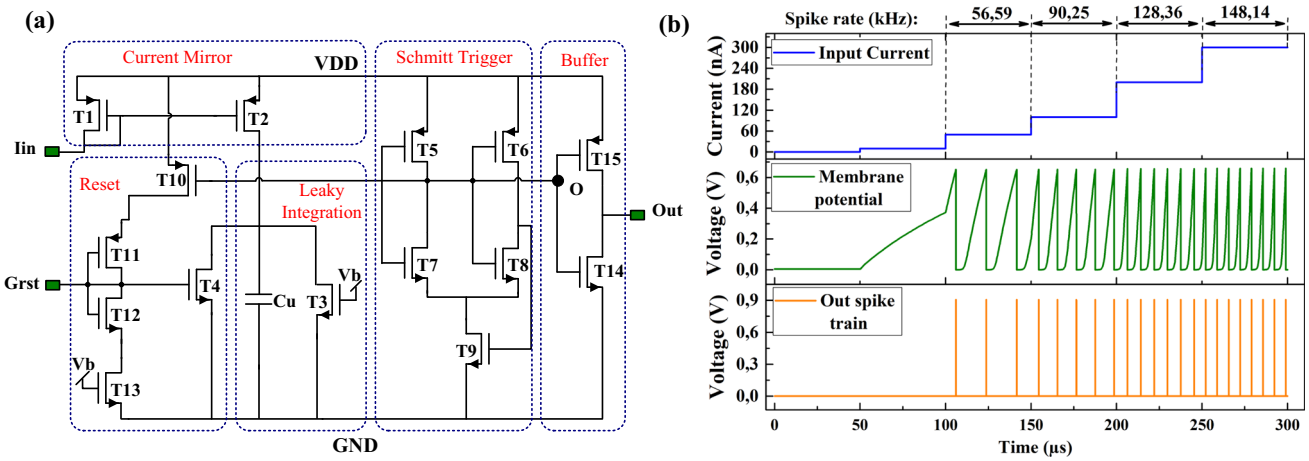

Fig. 4 **a** Functional circuit for the LIF neuron [30]. **b** Simulation results of the LIF neuron including the curves of the input current, the membrane potential, and the output spike train

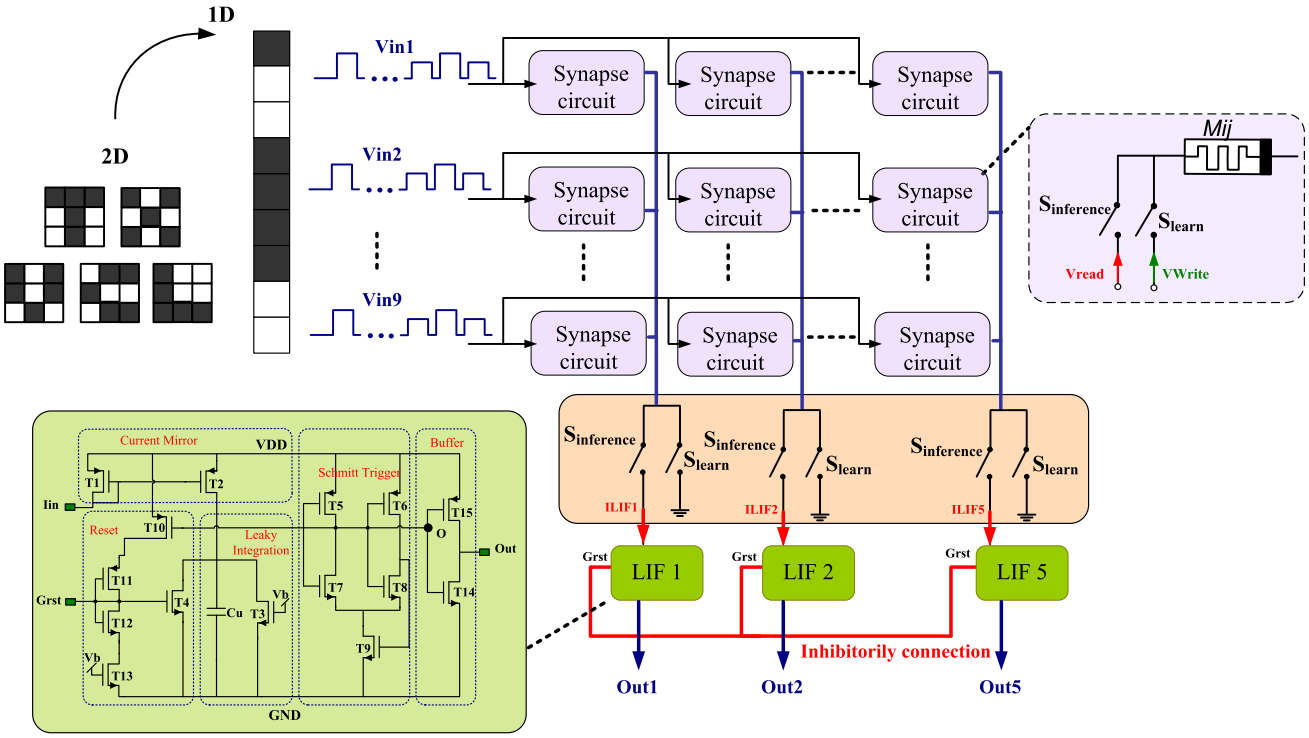

Fig. 5 Schematic of the simulated SNN network including input patterns, synaptic circuits and output Lif neurons

4.2 Simulation Results and Discussion

Due to the memristor model and the new programming technique, the memristances vary linearly, which greatly facilitates the weight adjustment process; i.e., an accurate weight value, can be set directly by controlling the voltage imposed on the synapse. Thus, during the learning phase, all synapses are programmed for a fixed period of 10 µs. Therefore, the synaptic weights are applied to the memristive crossbar according to the pixels of the training images (1 V for white pixels and 3 V for black pixels). In the inference phase, the pulse trains V_{in1} to V_{in9} are applied to the memristive crossbar encoding the five patterns to be classified. For example, the input voltages of image number 1 (pattern T) are: [$V_{11} = 0.7$ V, $V_{21} = 0.4$ V, $V_{31} = 0.4$ V, $V_{41} = 0.7$ V, $V_{51} = 0.7$ V, $V_{61} = 0.7$ V,$V_{71} = 0.7$ V, $V_{81} = 0.4$ V, $V_{91} = 0.4$ V].

The 9×5 crossbar efficiently executes the vector–matrix multiplication (VMM) of the input voltage by the conductance of the memristors and the summation of the resulting

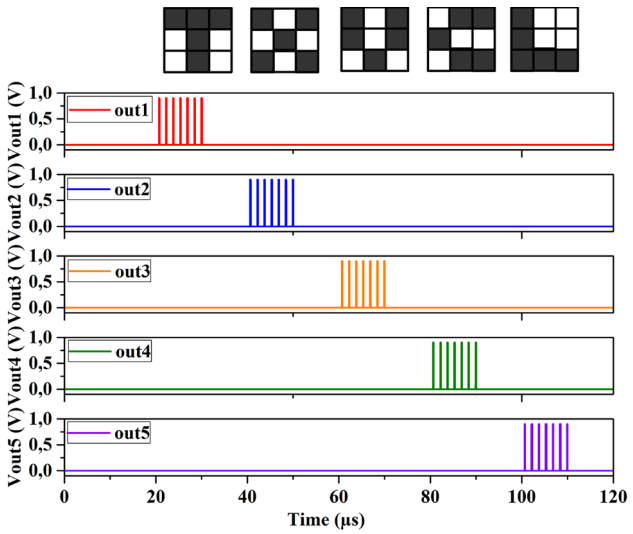

Fig. 6 Simulation results of the proposed SNN when applying training patterns

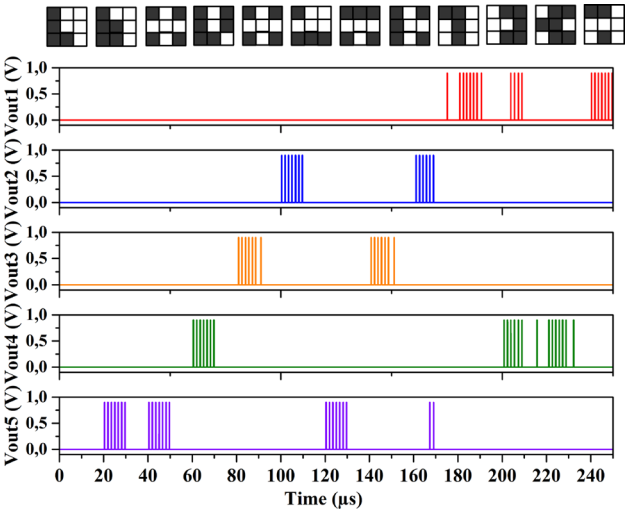

Fig. 7 Simulation results of the of the SNN when applying noisy patterns

currents over the five columns I_{LIF1} to I_{LIF5} of the crossbar, as given by the following expression:

$$I_{\mathrm{LIF}} = \sum_{i=0}^{9} G_{ij} V_i \tag{4}$$

Here, V_i represents the voltage applied to the i-th row of the crossbar and G_{ij} is the conductance of the memristors in the i-th row and j-th column. Our approach conducts the computation in a parallel mode. The inputs represented by analog voltage signals are sent to the crossbar array simultaneously. In this way, all the cells are selected and accessed at the same time. The impact of sneak path leakage is negligible in such a multiple inputs multiple output operation [31] so the current at each colon can follow Eq. (4).

Using the programming technique, the neurons in the output layer adapt their receptive fields according to training images. The first neuron generating a spike is the winner and will represent the implemented pattern. The output neurons are connected through the G_{rst} terminal to implement lateral inhibition.

The simulation results of the WTA network for the five trained patterns are shown in Fig. 6. Thus, as we excite the network with voltages corresponding to pattern number one, the neuron trained for "T" fires and causes all other neurons to maintain their internal state until the next pattern is entered. Therefore, all neurons are fired when the corresponding pattern is applied to the WTA. The T pattern is assigned to LIF_1, the X, V, C and L patterns are assigned to LIF_2, LIF_3, LIF_4 and LIF_5, respectively. We assign for the black pixel a voltage of 0.7 V, and for the white pixel a voltage of 0.4 V.

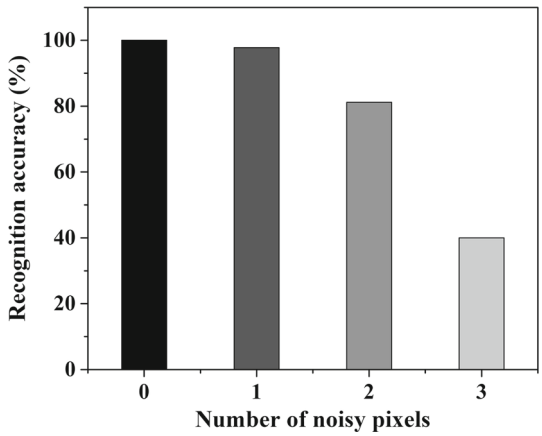

Fig. 8 Recognition accuracy as function of number of noisy pixels

These two voltages must be lower than the threshold voltages of the memristors to avoid changing the synaptic weights of the memristive crossbar.

The five images are given as input randomly several times and it is assessed whether they can be identified correctly or not. A perfect response of the overall system is exhibited. This system obtains a recognition performance of 100% and is capable of classifying 12 images in 300 µs, i.e., 40,000 images per second.

Furthermore, we test the system behavior when applied noisy patterns by flipping more than pixel which is demonstrated in Fig. 7. We confirm that our network can recognize noisy patterns.

The recognition accuracy obtained when applying "0", "1", "2" and "3" noisy pixels is 100%, 97.8%, 81.2% and 40%, respectively, as depicted in the histogram presented in Fig. 8. Hence, the accuracy decreases as the number of noisy pixels increases.

Table 2 Comparison of circuit implementation of different SNN memristive circuits

Memristive SNN circuits	Type of memristor	Synapse structure	Learning rule	Additional control units	Recognition rate	Output neuron
Le et al. [32]	RRAM device	2T1R	Widrow–Hoff algorithm learning	Microcontroller	Not given	Neuron model based on summing amplifier
Myonglae et al. [33]	PCMO-based device	1 M	STDP learning	Control unit	100%	LIF neuron
Yilong et al. [12]	RRAM device	1T1R	STDP learning	Control unit	78.9%	LIF neuron
Mirko et al. y[17]	Al/Al_2O_3/Nb_xO_y/Au-based device	1 M	STDP learning	Microcontroller	93.5%	Lif neuron
This work	EGM model	1 M	Programming technique	None	100%	LIF neuron

Moreover, our network demonstrates the robustness of the SNN circuit. The weight adjustment and the switching from learning phase to inference phase was validated with no additional control circuit.

In Table 2, we provide a comparison analysis of our work with some previous studies from literature that implement SNNs for recognition tasks.

Most of the adopted synapse structures in SNN [32] are based on 1T1R, 2T1R, 1 M. In our circuit we have used the most basic structure 1 M using the EGM model, which has promising results with smaller on-chip area.

[12, 17] and [33] implement STDP learning rule whereas ref. [32] is based on Widrow–Hoff algorithm learning. In this work, we tested and validated a new method to program the functioning of the SNN system which can be extended for other applications.

In in situ learning applied in these SNNs, the operation of the hardware circuit is inseparable from Microcontroller or FPGA.

In works [17] and [32], the weight adjustment and the network operation were controlled using a microcontroller.

In [12], a control unit was added to program the synapse. Then in [33] the hardware implementation of the neural network is based on FPGA circuit.

Thus, the use of Microcontroller or FPGA makes the circuit more complicated.

Besides, our proposed network is tolerable to noisy pattern and offer an efficient implementation in terms of area and accuracy.

5 Conclusion

In this paper, a spiking neural network has been validated using the WTA mechanism to perform an unsupervised on chip learning for recognition task. We approve a programming circuit implemented using the EGM model allowing read and write configurations.

In this WTA network, the output LIF neurons will compete with each other in order to identify the winner, that will actually indicate which of the learned patterns were displayed on the network input.

Based on our results, we prove not only an effective neural network circuit design but also an approach for combined both learning and inference on chip. The network has several relevant features, a good accuracy, small area and noise robustness. It can successfully classify 40,000 images per second.

The network can be extended into larger networks and offers efficient performance to accomplish more complex tasks. Further work should concentrate on optimizing multilevel conductance modulation for more useful applications.

References

1. Maas, W.: Networks of spiking neurons: the third generation of neural network models. Neural Netw. **10**(9), 1659–1671 (1997)
2. Merolla, P.A.; Arthur, J.V.; Alvarez-Icaza, R.; Cassidy, A.S.; Sawada, J.; Akopyan, F.; Jackson, B.L.; Imam, N.; Guo, C.; Nakamura, Y.; Brezzo, B.; Vo, I.; Esser, S.K.; Appuswamy, R.: A million spiking-neuron integrated circuit with a scalable communication network and interface. Science **345**, 668–673 (2014)
3. Cristini, A.; Salerno, M.; Susi, G.: A continuous-time spiking neural network paradigm. In: Advances in Neural Networks: Computational and Theoretical Issues, pp. 49–60 (2015)
4. Krestinskaya, O.; James, A.P.; Chua, L.O.: Neuromemristive circuits for edge computing: a review. IEEE Trans. Neural Netw. Learn. Syst. **31**(1), 4–23 (2020)
5. Yang, J.-Q.; Wang, R.; Ren, Y.; Mao, J.-Y.; Wang, Z.-P.; Zhou, Y.; Han, S.-T.: Neuromorphic engineering: from biological to spike-based hardware nervous systems. Adv. Mater. **32**(52), 2003610 (2020)
6. Chicca, E.; Indiveri, G.; Douglas, R.: An adaptive silicon synapse. In: Proceedings of the International Symposium on Circuits and Systems, ISCAS '03, Bangkok, Thailand (2003)
7. Seo, J.-S.; Brezzo, B.; Liu, Y.; Parker, B.D.; Esser, S.K.; Montoye, R.K.; Rajendran, B.; Tierno, J.A.; Chang, L.; Modha, D.S.; Friedman, D.J.: A 45 nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. In: IEEE Custom Integrated Circuits Conference (CICC), San Jose, CA, USA (2011)
8. Rziga, F.O.; Mbarek, K.; Ghedira, S.; Besbes, K.: The basic I-V characteristics of memristor model: simulation and analysis. Appl. Phys. A **123**(288), 1–8 (2017)
9. Lu, K.; Li, X.; Sun, Q.; Pang, X.; Chen, J.; Minari, T.; Liu, X.; Song, Y.: Solution-processed electronics for artificial synapses. Mater. Horizons **8**(2), 447–470 (2021)
10. Elhamdaoui, M.; Mbarek, K.; Ghedira, S.; Rziga, F.O.; Besbes, K.: Synapse design based on memristor. In: IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), Tunisia (2020)
11. Elhamdaoui, M.; Rziga, F.O.; Mbarek, K.; Besbes, K.: Spike-time-dependent plasticity rule in memristor models for circuit design. J. Comput. Electron. **21**, 1038–1047 (2022)
12. Guo, Y.; Wu, H.; Gao, B.; Qian, H.: Unsupervised learning on resistive memory array based spiking neural networks. Front. Neurosci. **13**, 812 (2019)
13. Kwon, M.-W.; Baek, M.-H.; Hwang, S.; Kim, S.; Park, B.-G.: spiking neural networks with unsupervised learning based on STDP using resistive synaptic devices and analog CMOS neuron circuit. Nanosci. Nanotechnol. **18**, 6588–6592 (2018)
14. Zhao, Z.; Qu, L.; Wang, L.; Deng, Q.; Li, N.; Kang, Z.; Guo, S.; Xu, W.: A memristor-based spiking neural network with high scalability and learning efficiency. IEEE Trans. Circuits Syst. II Express Briefs **67**(5), 931–935 (2020)
15. Saxena, V.; Wu, X.; Zhu, K.: Energy-efficient CMOS memristive synapses for mixed-signal neuromorphic system-on-a-chip. In: IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy (2018)
16. Yakopcic, C.; Hasan, R.; Taha, T.M.: Memristor based neuromorphic circuit for ex-situ training of multi-layer neural network algorithms. In: International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland (2015)
17. Hansen, M.; Zahari, F.; Kohlstedt, H.; Ziegler, M.: Unsupervised Hebbian learning experimentally realized with analogue memristive crossbar arrays. Sci. Rep. **8**(8914), 1–10 (2018)
18. Wang, J.J.; Yu, Q.; Hu, S.G.; Liu, Y.; Guo, R.; Chen, T.P.; Yin, Y.; Liu, Y.: Winner-takes-all mechanism realized by memristive neural network. Appl. Phys. Lett. **115**, 243701 (2019)
19. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S.: The missing memristor found. Nat. Lett. **453**, 80–83 (2008)
20. Yang, J.J.; Strukov, D.B.; Stewart, D.R.: Memristive devices for computing. Nat. Nanotechnol. **8**, 13–24 (2013)
21. Bi, G.-Q.; Poo, M.-M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, andpostsynaptic cell type. Neuroscience **18**(24), 10464–10472 (1998)
22. Pi, S.; Li, C.; Jiang, H.; Xia, W.; Xin, H.; Yang, J.J.; Xia, Q.: Memristor crossbars with 4.5 terabits-per-inch-square density and two nanometer dimension. Mesoscale and Nanoscale Physics (2018)
23. Zhao, M.; Gao, B.; Xi, Y.; Xu, F.; Wu, H.; Qian, I.: Endurance and retention degradation of intermediate levels in filamentary analog RRAM. J. IEEE de l'Electron Dev. Soc. **7**, 1239–1247 (2019)
24. Choi, B.J.; Torrezan, A.C.; Strachan, J.P.; Kotula, P.G.; Lohn, A.J.; Marinella, M.J.; Li, Z.; Williams, R.S.; Yang, J.J.: High-speed and low-energy nitride memristors. Adv. Funct. Mater. **26**(29), 5290–5296 (2016)
25. Indiveri, G.; Linares-Barranco, B.; Deligeorgis, R.L.G.; Prodromakis, T.: Integration of nanoscale memristor synapses in neuromorphic computing architectures. Nanotechnology **24**(38), 384010 (2013)
26. Linares-Barranco, B.: Memristors fire away. Nat. Electron. **1**, 100–101 (2018)
27. Rziga, F.O.; Mbarek, K.; Ghedira, S.; Besbes, K.: An efficient Verilog-A memristor model implementation: simulation and application. Comput. Electron. **18**(3), 1055–1064 (2019)
28. Yakopcic, C.; Taha, T.M.; Subramanyam, G.; Pino, R.E.: Generalized memristive device spice model and its application in circuit design. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **32**(8), 1201–1214 (2013)
29. Li, Y.; Zhong, Y.; Xu, L.; Zhang, J.; Xu, X.; Sun, H.; Miao, X.: Ultrafast synaptic events in a chalcogenide memristor. Sci. Rep. **3**(1619), 1–7 (2013)
30. Shamsi, J.; Mohammadi, K.; Shokouhi, S.B.: A low power circuit of a leaky integrate and fire neuron with global reset. In: Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran (2017)
31. Liu, C.; Yang, Q.; Yan, B.; Yang, J.; Du, X.; Zhu, W.; Jiang, H.; Wu, Q.; Barnell, M.; Li, H.: A memristor crossbar based computing engine optimized for high speed and accuracy. In: IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Pittsburgh, PA, USA (2016)
32. Yang, L.; Zeng, Z.; Shia, X.: A memristor-based neural network circuit with synchronous weight adjustment. Neurocomputing **363**(21), 114–124 (2019)
33. Chu, M.; Kim, B.; Park, S.; Hwang, H.; Jeon, M.; Lee, B.H.; Lee, B.-G.: Neuromorphic hardware system for visual pattern recognition with memristor array and CMOS neuron. IEEE Trans. Ind. Electron. **62**(4), 2410–2419 (2015)
34. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
35. Krestinskaya, O.; Ibrayev, T.; James, A.P.: Hierarchical temporal memory features with memristor logic circuits for pattern recognition. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **37**(6), 1143–1156 (2018)