A Memristor Crossbar-Based XNOR-Net Architecture for Image Recognition

Minh Le
HCMC University of Technology and Education, Ho Chi Minh City, Vietnam
leminh@hcmute.edu.vn

Son Ngoc Truong
HCMC University of Technology and Education, Ho Chi Minh City, Vietnam
sontn@hcmute.edu.vn (corresponding author)

Received: 27 April 2025 | Revised: 9 June 2025 | Accepted: 15 June 2025

 DOI: https://doi.org/10.48084/etasr.11761

ABSTRACT

XNOR-Net, employing the exclusive-NOR operation, has demonstrated a considerable enhancement in inference speed and a reduction in memory requirements for parameter storage. The hardware implementation of XNOR-Net has recently attracted much interest. This study proposes a memristor crossbar architecture to implement XNOR-Net for image recognition. The proposed memristor crossbar-based XNOR-Net consists of an input layer with 784 neurons, two hidden layers, each containing 500 neurons, and an output layer with 10 neurons for 10-digit recognition of MNIST images. The XNOR layer was implemented using a single memristor crossbar along with a constant current-generating circuit. This approach reduces the number of memristors by half, hence alleviating area and power overhead. In particular, in the output layer, all columns are fed into a Winner-Take-All (WTA) circuit to identify the column with the largest current. Therefore, the circuit that generates constant current can be excluded without affecting performance. In this way, the additional circuit can be removed, leading to a reduction in power consumption and area of the neuromorphic circuit. The memristor crossbar-based XNOR-Net showed a recognition rate of 94% on the MNIST dataset, producing correct outputs for input digits 0-9. The proposed memristor crossbar-based XNOR-Net demonstrates the potential for implementing memristor array-based deep network architectures in resource-constrained devices, especially in embedded and edge computing systems.

Keywords-XNOR-Net;memristor crossbar circuit ; neuromorphic computing system

I. INTRODUCTION

Deep Neural Networks (DNNs) have shown significant success in various areas such as computer vision, speech recognition, and natural language processing [1-3]. However, DNNs require massive computational and memory resources due to their deeply layered architecture, presenting challenges for their implementation in resource-constrained edge devices [4]. To mitigate these limitations, hardware-based DNN accelerators utilizing neuromorphic and in-memory computing paradigms have recently received significant attention [5, 6]. Although hardware-based DNNs present potential enhancements, executing high-precision Multiply-Accumulate (MAC) operations on hardware continues to be resource-intensive, thereby impacting chip density, energy efficiency, and processing speed. Binary Neural Networks (BNNs) have emerged as a viable alternative for edge AI applications by significantly reducing the complexity of neural computation [7-12]. In BNNs, synaptic weights and activations are limited to −1 or +1, denoting inhibitory and excitatory neural impulses, respectively [13, 14]. The multiplication of binary inputs and weights of -1 and +1 corresponds to an XNOR operation, resulting in the XNOR Network architecture (XNOR-Net), which has demonstrated applicability for embedded AI applications [11]. This simplification results in significant advantages in area, power consumption, and latency, making it very suitable for neuromorphic computing systems.

The hardware implementation of BNNs faces several challenges. Traditional CMOS-based neurons and synapses require a substantial amount of transistors to realize synaptic weights and arithmetic functions. Furthermore, the CMOS technology seems to be approaching its physical scaling constraints. Instead, memristor devices, conceptualized in 1971 and experimentally demonstrated in 2008, have become more attractive because of their advantages for neuromorphic applications, including nonvolatility, programmability, and nanoscale footprint [15-18]. Initial implementations, exemplified by the 1T1M (one transistor-one memristor) configuration, integrate memristors with CMOS transistors to create synaptic arrays. However, the presence of transistors amplifies both area and power consumption. On the contrary, pure memristor crossbars offer greater advantages in terms of power and area efficiency.

Numerous memristor crossbar-based neuromorphic systems have been exhibited in the past decade [17-27]. Previous studies have employed a dual memristor crossbar to implement the XNOR function at the circuit level [19, 20]. Two memristor crossbar arrays have been utilized to implement signed synaptic weights, [21, 24], but these designs may lead to the overhead of the memristor crossbar circuit. This work presents an innovative and optimized memristor crossbar architecture for the implementation of XNOR neural networks, utilizing -1 and +1 to represent both inputs and weights. The proposed architecture utilizes a single crossbar array and a constant current-generating circuit to perform the XNOR operations, in contrast to previous works that use two distinct crossbar arrays. This improvement decreases the required number of memristors by 50%, thus increasing the circuit density and bolstering overall system reliability. The proposed memristor crossbar-based XNOR-Net is applied to the cognitive task of image recognition using the MNIST dataset of handwritten digits. The XNOR-Net is trained offline on a server, and then the obtained weights are mapped onto memristor devices. The architecture is subsequently evaluated at the circuit level using Cadence Spectre circuit simulation. The proposed memristor crossbar-based circuit offers a promising architecture for scalable and energy-efficient neuromorphic hardware, making it highly appropriate for edge AI applications.

II. MEMRISTOR CROSSBAR-BASED XNOR NEURAL NETWORK

BNNs have demonstrated significant advantages for embedded applications and edge devices because of their reduced memory requirements and computational cost. Although BNNs usually exhibit lower recognition accuracy compared to their full-precision counterparts, this difference has been greatly reduced thanks to specialized training techniques designed for networks with binary weights [8-10]. When both the input and synaptic weights are constrained to values of -1 and 1, the multiplication operation is simplified to an XNOR operation. This architecture, known as XNOR-Net, can achieve improved accuracy while further reducing computational costs [11]. The XNOR-Net architecture is especially suitable for resource-constrained embedded systems used in edge computing applications. Figure 1 shows a concept of an XNOR-Net, where the inputs and weights are -1 and +1.

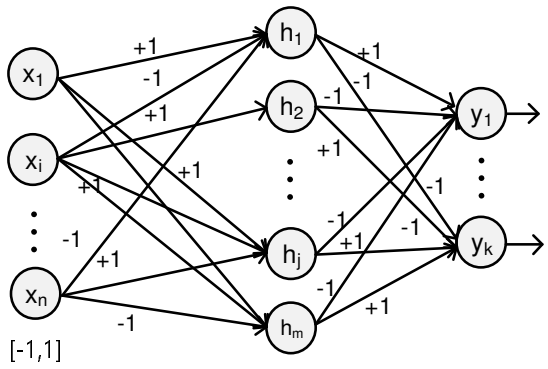

Fig. 1. The concept of a binary neural network, where both inputs and weights are constrained to -1 and +1.

Figure 1 shows a binary neural network with an input and weight of -1 and +1, consisting of an input layer, a hidden layer, and an output layer. Multiplying binary inputs and weights can be thought of as an XNOR operation. The following formula can be used to determine the output of the j^{th} neuron in the hidden layer:

$$h_j = \sum_{i=1}^{n} x_i w_{j,i} = \sum_{i=1}^{n} \overline{x_i \oplus w_{j,i}}$$

$$a_j = sign(h_j) = \begin{cases} -1 \; if \; h_i < 0 \\ +1 \; if \; h_j \geq 0 \end{cases} \quad (1)$$

where $w_{j,i}$ denotes the synaptic weight from the i^{th} input to the j^{th} neuron, $w_{j,i} \in \{-1,1\}$. Similarly, x_i represents the i^{th} input, which is also binarized to either -1 or +1. Under this constraint, the multiplication between x_i and $w_{j,i}$ is equivalent to an XNOR operation. h_j is fed into an activation function, implemented using a sign function, to produce the neuron's output a_j, which is also constrained to -1 or + 1.

The implementation of neural networks using memristor crossbar arrays offers significant advantages due to their inherent in-circuit computing capability. Specifically, vector-matrix multiplication and neural potential accumulation can be efficiently performed according to Ohm and Kirchhoff laws [17-23]. The XNOR operation can be expressed as a sum-of-products expression and realized through two complementary or twin crossbar arrays [19, 20]. This approach is generally applicable when both binary inputs and weights are constrained to values of 0 and 1. This work confirms that the expansion of XNOR remains valid when all inputs and weights are of -1 and +1, with -1 and +1 weights represented by HRS and LRS memristors, respectively. The XNOR operation can be expressed as a sum of products as follows:

$$h_j = \sum_{i=1}^{n} \overline{x_i \oplus w_{j,i}} = \sum_{i=1}^{n} \left(x_i w_{j,i} + x_i' w_{j,i}' \right) \quad (2)$$

This equation presents an expansion of the XNOR function to a sum of products, where x_i' is an inversion of the i^{th} input and $w_{j,i}'$ is the inversion of weights $w_{j,i}$. Figure 2 shows the crossbar circuit that implements the expansion of XNOR [19].

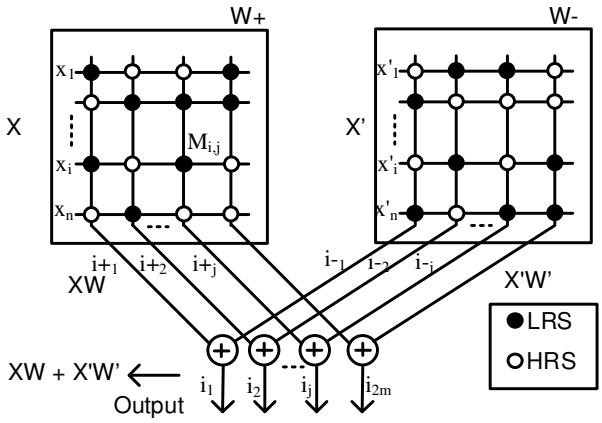

Fig. 2. The crossbar circuit to implement the vector-matrix XNOR operation.

The output currents i_1 to i_m denote neuron outputs derived from the XNOR operation between the input vector and the weight matrix. The memristor-based circuit, illustrated in Figure 2, has proven effective when both the inputs and weights have binary values of 0 and 1, with weight values of 0 and 1 represented by High Resistance State (HRS) and Low Resistance State (LRS) memristors, respectively [19, 20]. In neuromorphic computing, synaptic weights are generally denoted by both negative and positive values, reflecting inhibitory and excitatory synapses, respectively. Therefore, the binary representations of -1 and +1 are utilized more frequently than 0 and 1 in binary neural networks. This work utilized the memristor crossbar array to perform the XNOR operation, with both the input vector and the weight matrix being in binary values of -1 and +1. It must be verified that the memristor crossbar accurately generates XNOR operation outputs under this encoding scheme, similar to its performance with binary inputs and weights of 0 and 1.

Table I shows the output current with various combinations of input and weight values. The input values of -1 and +1 are represented by voltages of -1 V and +1 V, respectively. The weight values of -1 and +1 are represented by HRS and LRS memristors, respectively. Assuming that the HRS value is much greater than the LRS value, 1/HRS can therefore be considered approximately zero. The resulting column current is positive when the input and the weight are identical, and negative when they are dissimilar, as illustrated in Table I. This verifies that the complementary memristor crossbar array properly executes the XNOR function when both inputs and weights are expressed with bipolar values of -1 and +1.

TABLE I. OUTPUT COLUMN CURRENTS WITH VARIOUS COMBINATIONS OF INPUT AND WEIGHT

x_i	$w_{j,i}$ (memristance)	$x_i w_{j,i}$	$x_i' w_{j,i}'$	$(x_i w_{j,i} + x_i' w_{j,i}')$
-1	-1 (HRS)	-1/HRS	+1/LRS	+1/LRS
-1	+1 (LRS)	-1/LRS	+1/HRS	-1/LRS
+1	-1 (HRS)	+1/HRS	-1/LRS	-1/LRS
+1	+1 (LRS)	+1/LRS	-1/HRS	+1/LRS

Memristor crossbar arrays have shown outstanding performance in neuromorphic computing systems. The fabrication of large-scale memristor crossbars has many challenges. Reducing the number of memristors in a neuromorphic circuit is crucial for improving performance and feasibility. This work presents a crossbar architecture that uses a singular crossbar array along with a constant current-generating circuit to perform the XNOR function, rather than employing two separate crossbar arrays. The concept derives from the expansion of the XNOR function as detailed below:

$$\begin{aligned} h_j &= \sum_{i=1}^{n} \overline{x_i \oplus w_{j,i}} = \sum_{i=1}^{n} \left(x_i w_{j,i} + x_i' w_{j,i}'\right) \\ &= \sum_{i=1}^{n} \left(x_i w_{j,i} + x_i' \left(1 - w_{j,i}\right)\right) \\ &= \sum_{i=1}^{n} \left((x_i - x_i') w_{j,i} + x_i' . 1\right) \\ &= 2 \sum_{i=1}^{n} \left(x_i w_{j,i} + \frac{1}{2} x_i' . 1\right) \end{aligned} \quad (3)$$

This equation demonstrates that the vector-matrix XNOR can be performed using a single crossbar plus an additional inverted term input through a weight of 1. The term $x_i' . 1$ is added to all columns, functioning as a constant current across each column. To validate (3) accurately with both inputs and weights constrained to -1 and +1, the column current is calculated using various combinations of input and weight, as presented in Table II.

TABLE II. THE OUTPUT COLUMN CURRENT OF SINGLE CROSSBAR AND CONSTANT CURRENT-GENERATING CIRCUIT WITH VARIOUS COMBINATIONS OF INPUT AND WEIGHT

x_i	$w_{j,i}$ (memristor)	$x_i w_{j,i}$	$\frac{1}{2} x_i'$	h_j
-1	-1 (HRS)	-1/HRS	+1/2LRS	+1/LRS
-1	+1 (LRS)	-1/LRS	+1/2LRS	-1/LRS
+1	-1 (HRS)	+1/HRS	-1/2LRS	-1/LRS
+1	+1 (LRS)	+1/LRS	-1/2LRS	+1/LRS

Table II shows the tested values of the input and weight to evaluate the performance of the XNOR operation according to (3), with the input and weight set to binary values of -1 and +1. The column current derived from (3) is identical to that obtained from (2), as seen in Tables I and II. An interesting discovery is that the vector-matrix XNOR can be implemented on a memristor crossbar integrated with a circuit that provides constant current to all columns.

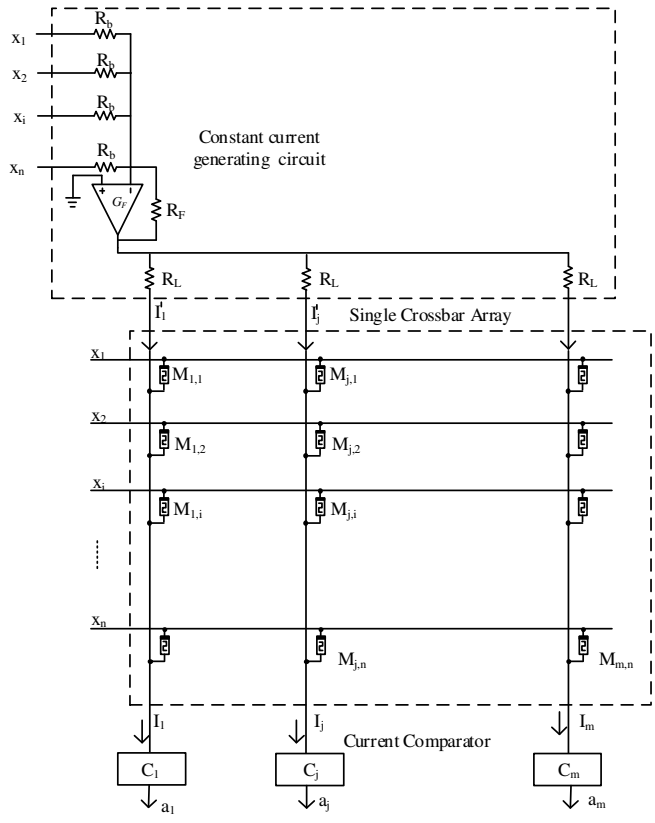

Fig. 3. The memristor crossbar circuit employs a crossbar array and a constant current-generating circuit to implement the vector-matrix XNOR operation in memristor-base XNOR-Net.

Figure 3 shows a memristor crossbar architecture, which utilizes a single crossbar array in conjunction with a constant current-generating circuit to perform the XNOR function within a layer of XNOR-Net. Binary inputs of -1 and +1 are applied to the crossbar array to produce column currents. Concurrently, similar inputs are supplied to an inverting operational amplifier (op-amp) circuit to generate a consistent constant current across all columns. Let $w_{j,i}$ represent the synaptic weight associated with the i^{th} input to the j^{th} neuron. The constant current produced by the circuit is introduced into each column and can be articulated as:

$$I_j' = \frac{R_F}{R_L R_b}\sum_{i=1}^{n}(-x_i)$$

$$I_j' = \frac{1}{2}\sum_{i=1}^{n}(x_i')/LRS \quad (4)$$

where $R_L = 2R_F$ and $R_b = LRS$. Under this condition, the resulting current I_j' aligns with the expression in (3). In (3), 1 means the weight of +1 and is represented by a current of $1/LRS$. This current I_j' is then injected into the j^{th} column, and the total current can be as calculated as:

$$I_j = \sum_{i=1}^{n}\left(\frac{x_j}{M_{j,i}} + \frac{1}{2}x_i'.LRS\right)$$

$$I_j = \sum_{i=1}^{n}\left(x_j w_{j,i} + \frac{1}{2}x_i'.1\right) \quad (5)$$

where $w_{j,i}$ is represented by $1/M_{j,i}$, and the weight of 1 is represented by the value of 1/LRS. I_j is the total current of the j^{th} column, corresponding to the output of the j^{th} neuron, h_j. The resulting current is subsequently passed through a sign function C_j, which outputs +1 if the column current is positive, and -1 otherwise, implemented by a current comparator using a simple CMOS circuit [28]. Therefore, the coefficient of 2 in (3) can be omitted without affecting functionality.

From (3), an interesting property can be observed; in the expansion of vector-matrix XNOR operation, the identical term $\frac{1}{2}x_i'.1$ is added to all columns. Consequently, removing this term results in an equal loss of current across all columns. This reduction does not affect the task of identifying the column with the maximum current. Therefore, when using a Winner-Take-All (WTA) circuit to determine the column with the highest current, the constant term $\frac{1}{2}x_i'.1$ can be omitted. As a result, the constant current-generating circuit in the output layer can be eliminated.

Figure 4 shows the design of the proposed memristor crossbar architecture to implement XNOR-Net. This architecture was applied to the cognitive task of image recognition using the MNIST dataset of handwritten digits. Each image has a size of 28×28 and is flattened into a vector of 784 elements. The network architecture has feedforward layers with 784 input neurons, two hidden layers of 500 neurons each, and an output layer with 10 output neurons corresponding to the ten-digit classes. The connection between the input and the first hidden layers is implemented using a single crossbar array paired with a constant current-generating circuit. Column currents from the crossbar are directed into the current comparators, which perform the activation function. The output of the comparators, functioning as sign activations, are subsequently used as the inputs for the second hidden layer. The column currents of the second crossbar are passed into the current comparators to produce the activated outputs, as illustrated in Figure 4. In the hidden-to-output layer, the total column currents enter a winner-take-circuit to identify the column with the highest current, hence the constant current generating circuit can be omitted, as previously mentioned. The output o_1 to o_{10} signify the neuron's responses for the recognition of ten digits.

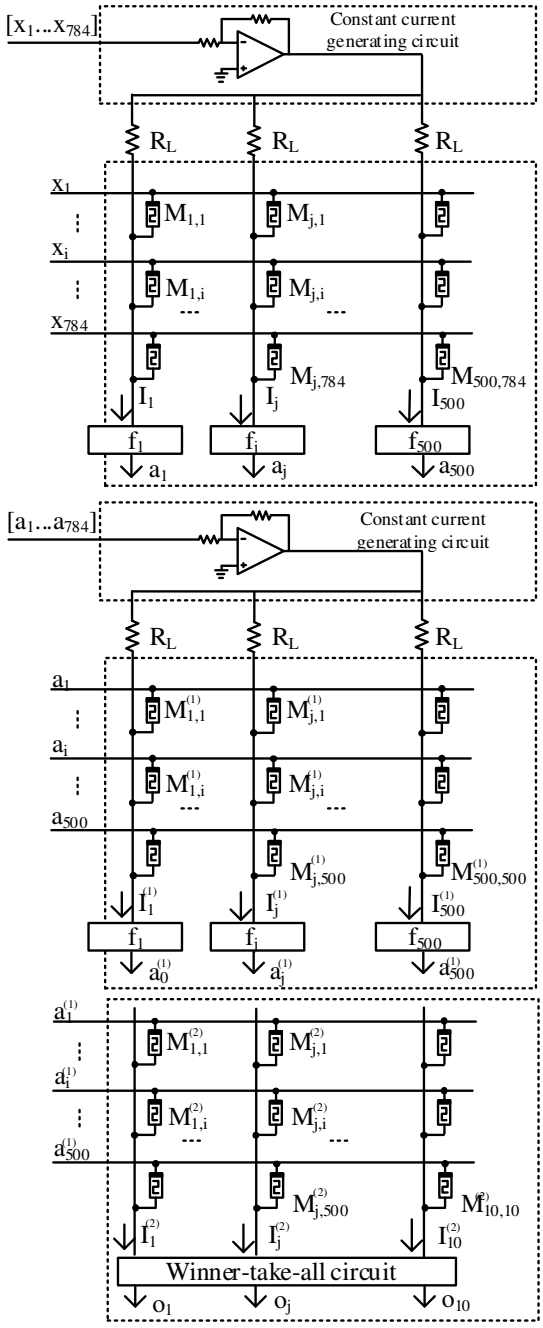

Fig. 4. The schematic of the proposed memristor-based XNOR neural network for handwritten digit recognition.

III. SIMULATION RESULTS AND DISCUSSIONS

The MNIST dataset, a widely used benchmark for evaluating image recognition models [29], was employed to evaluate the proposed memristor crossbar-based XNOR. The dataset comprises 60,000 training images and 10,000 test images of greyscale handwritten digits, each with a resolution of 28×28 pixel. For compatibility with the XNOR-Net architecture, the grayscale images were binarized into black-and-white images and represented as vectors with values of -1 and +1. The XNOR-Net was trained offline using the Straight-Through Estimator (STE) method, which has shown efficiency for neural networks when both inputs and weights are binary [30]. The weights obtained are subsequently mapped to the memristor devices, where binary values of -1 and +1 are represented by the HRS and LRS of memristors, respectively. The proposed memristor crossbar-based XNOR-Net was simulated in a Cadence Spectre circuit simulation. The memristor was modeled using Verilog-A, which has been validated for accurately emulating the behavior of physical memristor devices [31]. The input values of -1 and +1 correspond to voltage levels of -1 V and +1 V, respectively. This voltage level prevents memristance drift during the inference phase. A custom Verilog-A framework was developed to sequentially input data and read the output of the WTA circuit to evaluate the recognition rate. The simulation results demonstrate that the proposed memristor crossbar-based XNOR-Net achieved a recognition rate of 94% on the MNIST dataset.

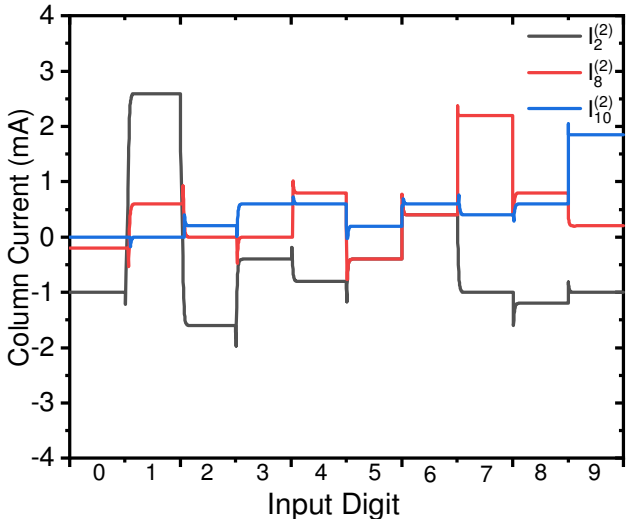

Fig. 5. Accumulated current of columns 1, 7, and 9 when the digits 0–9 are sequentially applied to the XNOR-Net.

Figure 5 shows the simulation results of the proposed memristor crossbar-based XNOR-Net circuit. The column current of the output layer was measured before the WTA circuit, while digits 0-9 were sequentially applied to the inputs. As illustrated in Figure 5, the current of column 1 (black line) increases significantly when the input digit 1 is applied, indicating that digit 1 was successfully recognized. Similarly, the current of column 7, which is represented by the red line, increases when the input digit 7 is applied, and the current of column 9, which is represented by the blue line, is activated with the input digit 9. These results confirm the proper operation of the proposed memristor crossbar-based XNOR-Net.

The XNOR-Net, employing the exclusive-NOR operation, has demonstrated a significant enhancement in inference speed and a reduction in memory requirements for parameter storage. The hardware implementation of XNOR-Net has recently garnered significant interest. Discovering new technologies as alternatives for the implementation of CMOS neural networks provides significant advantages regarding power consumption, area efficiency, and scalability for neuromorphic systems, especially in edge computing applications. To construct the XNOR function and signed weights, previous studies used two memristor crossbar arrays [19-21]. The use of two arrays leads to high area and power overhead, along with increased complexity in the fabrication and programming of the memristor arrays. This work presents a potential memristor architecture to implement XNOR-Net. The proposed architecture uses a single memristor array in conjunction with a constant current-generating circuit. This is accomplished by elaborating the expression of the XNOR function and implementing it at the circuit level. Another contribution is the elimination of the constant current-generating circuit at the output layer, which allows the use of a single memristor crossbar array without the need for extra circuitry when paired with a WTA circuit. Consequently, both the circuit area and power consumption exhibit substantial enhancement. The proposed XNOR-Net demonstrates the potential to implement memristor crossbar-based deep learning architecture on resource-limited devices, particularly in embedded systems and edge computing applications.

IV. CONCLUSION

This work presents a memristor crossbar circuit for the implementation of an XNOR neural network for image recognition. The proposed memristor crossbar architecture employs a single memristor crossbar and a constant current-generating circuit to perform the exclusive NOR operation, thus saving 50% of the amount of memristors, compared to utilizing two memristor crossbars. The memristor crossbar-based XNOR-Net demonstrated a recognition rate of 94% for the MNIST dataset. The simulation results demonstrate that the memristor crossbar-based XNOR-Net circuit generates the right output for digits 0-9. The proposed XNOR network exhibits the potential to deploy memristor crossbar-based deep network architectures on resource-constrained devices, particularly in embedded systems and edge computing applications.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, Feb. 2017, https://doi.org/10.1145/3065386.

[2] S. R. Shahamiri, "Speech Vision: An End-to-End Deep Learning-Based Dysarthric Automatic Speech Recognition System," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 852–861, 2021, https://doi.org/10.1109/TNSRE.2021.3076778.

[3] D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," *IEEE Transactions on*

Neural Networks and Learning Systems, vol. 32, no. 2, pp. 604–624, Feb. 2021, https://doi.org/10.1109/TNNLS.2020.2979670.

[4] J. Lee, J. Lee, D. Han, J. Lee, G. Park, and H. J. Yoo, "An Energy-Efficient Sparse Deep-Neural-Network Learning Accelerator With Fine-Grained Mixed Precision of FP8–FP16," *IEEE Solid-State Circuits Letters*, vol. 2, no. 11, pp. 232–235, Nov. 2019, https://doi.org/10.1109/LSSC.2019.2937440.

[5] J. Y. Huang, J. L. Syu, Y. T. Tsou, S. Y. Kuo, and C. R. Chang, "In-Memory Computing Architecture for a Convolutional Neural Network Based on Spin Orbit Torque MRAM," *Electronics*, vol. 11, no. 8, Apr. 2022, Art. no. 1245, https://doi.org/10.3390/electronics11081245.

[6] M. Cho and Y. Kim, "FPGA-Based Convolutional Neural Network Accelerator with Resource-Optimized Approximate Multiply-Accumulate Unit," *Electronics*, vol. 10, no. 22, Nov. 2021, Art. no. 2859, https://doi.org/10.3390/electronics10222859.

[7] Y. Wang, J. Lin, and Z. Wang, "An Energy-Efficient Architecture for Binary Weight Convolutional Neural Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 2, pp. 280–293, Feb. 2018, https://doi.org/10.1109/TVLSI.2017.2767624.

[8] T. Simons and D. J. Lee, "A Review of Binarized Neural Networks," *Electronics*, vol. 8, no. 6, Jun. 2019, Art. no. 661, https://doi.org/10.3390/electronics8060661.

[9] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1." arXiv, Mar. 17, 2016, https://doi.org/10.48550/arXiv.1602.02830.

[10] C. Baldassi, A. Braunstein, N. Brunel, and R. Zecchina, "Efficient supervised learning in networks with binary synapses," *Proceedings of the National Academy of Sciences*, vol. 104, no. 26, pp. 11079–11084, Jun. 2007, https://doi.org/10.1073/pnas.0700324104.

[11] M. Rastegari, V. Ordonez, J. Redm on, and A. Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks." arXiv, Aug. 02, 2016, https://doi.org/10.48550/arXiv.1603.05279.

[12] S. N. Truong, "A Low-cost Artificial Neural Network Model for Raspberry Pi," *Engineering, Technology & Applied Science Research*, vol. 10, no. 2, pp. 5466–5469, Apr. 2020, https://doi.org/10.48084/etasr.3357.

[13] L. F. Abbott and W. G. Regehr, "Synaptic computation," *Nature*, vol. 431, no. 7010, pp. 796–803, Oct. 2004, https://doi.org/10.1038/nature03010.

[14] R. S. Zucker and W. G. Regehr, "Short-Term Synaptic Plasticity," *Annual Review of Physiology*, vol. 64, no. 1, pp. 355–405, Mar. 2002, https://doi.org/10.1146/annurev.physiol.64.092501.114547.

[15] L. Chua, "Memristor-The missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971, https://doi.org/10.1109/TCT.1971.1083337.

[16] R. Williams, "How We Found The Missing Memristor," *IEEE Spectrum*, vol. 45, no. 12, pp. 28–35, Dec. 2008, https://doi.org/10.1109/MSPEC.2008.4687366.

[17] A. Gebregiorgis *et al.*, "Tutorial on memristor-based computing for smart edge applications," *Memories - Materials, Devices, Circuits and Systems*, vol. 4, Jul. 2023, Art. no. 100025, https://doi.org/10.1016/j.memori.2023.100025.

[18] M. Kimura, R. Tanaka, S. Akane, I. Horiuchi, Y. Hiroshima, and Y. Nakashima, "Neuromorphic System Using Crosspoint-Type TaO_x/Ta Memristors and Direct Device Training for Associative Memory," *IEEE Transactions on Electron Devices*, vol. 70, no. 9, pp. 4635–4640, Sep. 2023, https://doi.org/10.1109/TED.2023.3296393.

[19] S. N. Truong, S. J. Ham, and K. S. Min, "Neuromorphic crossbar circuit with nanoscale filamentary-switching binary memristors for speech recognition," *Nanoscale Research Letters*, vol. 9, no. 1, Dec. 2014, Art. no. 629, https://doi.org/10.1186/1556-276X-9-629.

[20] S. N. Truong, S. Shin, S. D. Byeon, J. Song, and K. S. Min, "New Twin Crossbar Architecture of Binary Memristors for Low-Power Image Recognition With Discrete Cosine Transform," *IEEE Transactions on Nanotechnology*, vol. 14, no. 6, pp. 1104–1111, Nov. 2015, https://doi.org/10.1109/TNANO.2015.2473666.

[21] F. Aguirre *et al.*, "Hardware implementation of memristor-based artificial neural networks," *Nature Communications*, vol. 15, no. 1, Mar. 2024, Art. no. 1974, https://doi.org/10.1038/s41467-024-45670-9.

[22] G. W. Burr *et al.*, "Neuromorphic computing using non-volatile memory," *Advances in Physics: X*, vol. 2, no. 1, pp. 89–124, Jan. 2017, https://doi.org/10.1080/23746149.2016.1259585.

[23] S. Wen *et al.*, "Memristor-Based Design of Sparse Compact Convolutional Neural Network," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1431–1440, Jul. 2020, https://doi.org/10.1109/TNSE.2019.2934357.

[24] G. W. Burr *et al.*, "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element," in *2014 IEEE International Electron Devices Meeting*, San Francisco, CA, USA, Dec. 2014, https://doi.org/10.1109/IEDM.2014.7047135.

[25] S. S. Archita and V. Ravi, "Review of memristor based neuromorphic computation: opportunities, challenges and applications," *Engineering Research Express*, vol. 6, no. 3, Sep. 2024, Art. no. 032203, https://doi.org/10.1088/2631-8695/ad6662.

[26] X. Wu, B. Dang, H. Wang, X. Wu, and Y. Yang, "Spike-Enabled Audio Learning in Multilevel Synaptic Memristor Array-Based Spiking Neural Network," *Advanced Intelligent Systems*, vol. 4, no. 3, Mar. 2022, Art. no. 2100151, https://doi.org/10.1002/aisy.202100151.

[27] Y. Yu, H. Tang, X. Feng, X. Wang, and H. Huang, "Design of multilayer cellular neural network based on memristor crossbar and its application to edge detection," *Journal of Systems Engineering and Electronics*, vol. 34, no. 3, pp. 641–649, Jun. 2023, https://doi.org/10.23919/JSEE.2022.000127.

[28] H. Träff, "Novel approach to high speed CMOS current comparators," *Electronics Letters*, vol. 28, no. 3, pp. 310–311, Jan. 1992, https://doi.org/10.1049/el:19920192.

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, https://doi.org/10.1109/5.726791.

[30] Y. Bengio, N. Léonard, and A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation." arXiv, Aug. 15, 2013, https://doi.org/10.48550/arXiv.1308.3432.

[31] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, "A Memristor Device Model," *IEEE Electron Device Letters*, vol. 32, no. 10, Oct. 2011, https://doi.org/10.1109/LED.2011.2163292.

AUTHORS PROFILE

Minh Le received a B.S. in electrical and electronic engineering and an M.S in electronic engineering, Ho Chi Minh City University of Technology and Education, Vietnam, in 2009 and 2012, respectively. He is currently working toward a Ph.D. degree at HCMC University of Technology and Education, Vietnam. His research interests are neuromorphic computing and brain-inspired systems.

Son Ngoc Truong received his B. S. and M.S. degrees in Electronic Engineering from Ho Chi Minh City University of Technology and Education, Vietnam, in 2006 and 2011, respectively. He earned his Ph.D. in Electronic Engineering at Kookmin University, Seoul, South Korea, in 2016. He was a postdoctoral researcher at Kookmin University from 2016 to 2017. He is currently an Associate Professor at Ho Chi Minh City University of Technology and Education, Vietnam. His research interests include neuromorphic computing systems, brain-inspired circuits, hardware acceleration of deep neural networks, and related areas of electronic systems design.