Journal of
Low Power Electronics
Vol. 15, 214–223, 2019

Efficient and Low Overhead Memristive Activation Circuit for Deep Learning Neural Networks

Anu Bala*, Xiaohan Yang, Adedotun Adeyemo, and Abusaleh Jabir

School of Engineering, Computing and Mathematics, Oxford, OX33 1HX, UK

(Received: 1 March 2019; Accepted: 27 March 2019)

An efficient memristor MIN function based activation circuit is presented for memristive neuromorphic systems, using only two memristors and a comparator. The ReLU activation function is approximated using this circuit. The ReLU activation function helps to significantly reduce the time and computational cost of training in neuromorphic systems due to its simplicity and effectiveness in deep neural networks. A multilayer neural network is simulated using this activation circuit in addition to traditional memristor crossbar arrays. The results illustrate that the proposed circuit is able to perform training effectively with significant savings in time and area in memristor crossbar based neural networks.

Keywords: Neural Network, MIN-MAX Function, Memristor Crossbar Array, ReLU.

1. INTRODUCTION

Memristors[1] are becoming an active and timely topic in neuromorphic systems due to their diverse and effective properties, such as non-volatility, low power consumption as well as nanoscale size.[2,3] Designing a neural network with a higher synaptic density is a major obstacle faced by the existing technology. This can be elegantly solved with neural networks designed with memristors due to their unique physical layout.[4] Hence, the research suggests that the physical memristors are capable of generating high density and low power hardware neural systems, which will perform fast and effective dot product operations in parallel.

A number of memristor fabrication materials and devices have been introduced subsequently. A new vision and research activities in the applications of memory[5] and neuromorphic domain[6] have been raised due to the foundation of this mathematical device. The resistance of a memristor device can be altered by applying voltage pulses same as the spikes that are applied to change the weight of a biological synapse. Thus, this nanoscale device has the potential to behave as a biological synapse.[7,8]

Memristive crossbar circuits are used to develop neuromorphic systems in Refs. [9, 10] which illustrates the ability of a memristor crossbar structure in the implementation of high density networks. The research work in Refs. [11, 12] experimentally demonstrates the fully operational artificial neural network and effective implementation of the locally competitive algorithm (LCA) for feature extraction based on integrated memristor crossbar arrays. Moreover, its low power consumption and nanoscale characteristics make it suitable for storage and memory organization in the field of image processing.[13] It also shows the potential to use in analog circuit[14] and logic design[15] applications.

In a neural network, an activation function introduces non-linearity into the network which helps the network with learning and performing complex functions. Thus, an activation function is an essential part of an artificial neural network. The widely used activation functions are sigmoid, tanh and rectified linear unit (ReLU).[2,3,16–18] The ReLU activation function, introduced very recently,[19] is becoming more popular in deep neural networks due to its simplicity. It also provides sparsity and aids to eliminate the gradient problem that is hard to handle with sigmoid and tangent functions. Moreover, it has been shown that deep networks can be trained efficiently using ReLU even without pre-training.[16] Most of the activation functions used in memristive neural networks in literature are approximated as sigmoid, tanh and piecewise linear function.[3,17,18] The majority of these functions are implemented using only operational amplifiers, which incur significant hardware overhead. In contrast, considering the merits and demerits of the existing approaches, this paper presents a novel implementation of the ReLU activation function based on memristor MIN functionality. To the best of our

*Author to whom correspondence should be addressed.
Email: 15057719@brookes.ac.uk

 1546-1998/2019/15/214/010 doi:10.1166/jolpe.2019.1600

knowledge, this is the first circuit which approximates an activation function using memristors. We show with experimental results that the proposed circuit is both significant area efficient as well as aids very fast learning.

The rest of the paper is organized as follows: The basic memristor crossbar structure and MIN functionality are reviewed in Section 2. It also gives the overview about memristor device and neural network. Section 3 describes the proposed memristive activation circuit and the experimental details of the network simulation. It also explains the method of reading and writing a memristor in the memristor crossbar. Section 4 illustrates memristor crossbar architectures used for implementing different multilayer neural networks and experimental setup. Section 5 evaluates the results. Finally, the paper is concluded in Section 6.

2. BACKGROUND

2.1. Memristive Device

Memristor is a two-terminal non-volatile passive element with varying resistance and pinched hysteresis loop is the recognising feature of a memristor. This fourth basic passive element was first theorized by Professor Leon Chua in 1971[1] and it predicts the relationship between charge and magnetic flux. The physical memristor device was fabricated in 2008 by HP Labs[4] based on TiO_2 thin film structure. It consists of two thin films sandwiched between the pair of platinum electrodes as shown in Figure 1(a). It contains two regions named as doped and undoped region. Undoped region contains titanium dioxide TiO_2 and doped region contains poor-oxygen titanium dioxide TiO_{2-x}. Undoped region contains more oxygen vacancies than doped region and has high resistance whereas doped region acts as semiconductor due to positive dopants and has low resistance. Both doped and undoped regions are modeled as resistors and it is equivalent to two resistors in series as shown in Figure 1(b). Memristor switching is based on the voltage and current pass through it. Its resistance changes from high resistance state (HRS) R_{off} to low resistance state (LRS) R_{on} and vice-versa based on the bias of voltage applied to its terminals. Ohm's law is used to describe the relationship between current and voltage of a memristor as shown in Eq. (1). In Figure 1(a), D represents the total length of the device (doped and undoped region) and w represents the length of the doped region.

$$V(t) = M(x)i(t) \tag{1}$$

$$M(x) = R_{on}(x) + R_{off}(1-x) \tag{2}$$

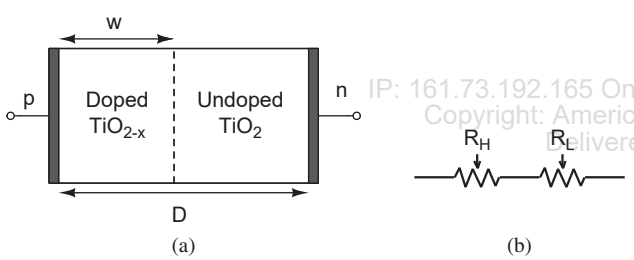

Fig. 1. (a) TiO_2 memristor model structure (b) equivalent circuit.

2.2. Artificial Neural Networks and Memristor Crossbar

The human brain is the inspiration of artificial neural network. Neurons and synapses are two basic elements of any neural network. In Artificial Neural Networks (ANNs), hundreds of neurons are connected with synapses and organized in single or multiple layers. Each synapse contains the weight parameter. Each neuron constitutes the

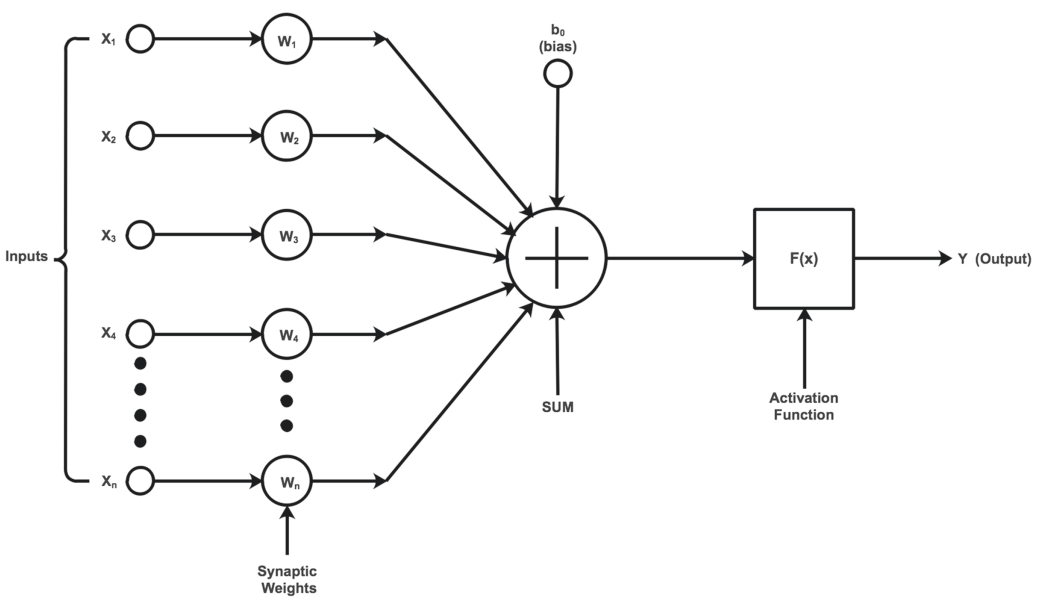

Fig. 2. Basic structure of neural network.

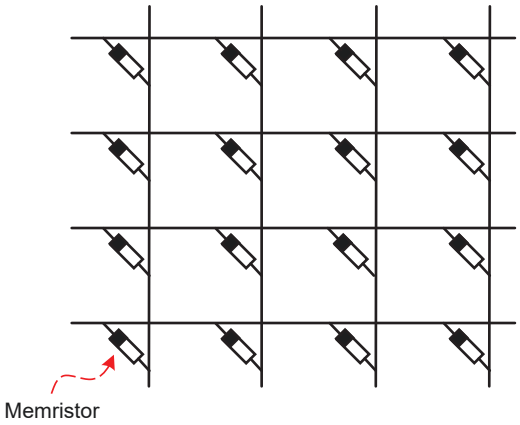

Fig. 3. Memristor crossbar circuit.

weighted sum of inputs and passes the information to the threshold function as represented by Eq. (3). The threshold function provides non-linearity to the network. Figure 2 shows a basic structure of a neural network. Weight parameters can be adjusted according to learning rule during training process. Numerous neural network structures have been developed for pattern recognition, character recognition, control signal and so on. Feed forward neural networks, Convolutional neural networks, and recurrent hopfield neural networks are some of the examples of ANNs.

$$Y_j = F\left(\sum_i X_i W_{i,j} + b\right) \quad (3)$$

X_i and $W_{i,j}$ represent input and weight for each neuron and b is the bias in the network.

Memristor crossbars have the potential to gain high density, which is desired in memristor-based memories and neuromorphic systems. Memristor crossbar architecture, comprises a horizontal and a vertical set of bars that lies perpendicular to on each other. A memristor is positioned between each horizontal and vertical intersection as shown in Figure 3. High density neural networks can be built using memristor crossbar due to its incredible characteristics of weight storage and vector-matrix multiplication.[2,10,17] Each horizontal and vertical line in a crossbar employs a memristor that is used as a synapse. The Rows and the columns are represented as inputs and outputs respectively. Memristor crossbar yields the multiplication of the inputs and weights. The Voltage at each row is considered as input and the conductance of each memristor is considered as weight. The dot product of each column in memristor crossbar is given by:

$$V_{\text{out}_j} = \frac{\sum_i V_{\text{in}_i} C_{i,j}}{\sum_i C_{i,j}} \quad (4)$$

V_{in_i} and $C_{i,j}$ denote voltage and conductance for each input i in memristor crossbar structures.

Equation (4) depicts the output at each column. Two columns are used to identify the output of a single neuron at each column. The difference between two column output is considered as the final output of each neuron in memristor crossbars. Equation (4) is defined by a voltage divider between the memristor conductances (i.e., the reciprocal of resistance) in crossbar structure.

2.3. MIN-MAX Function

Figure 4(a) represents a diagram of a single memristor, where it is assumed that when the voltage difference between the p terminal and the n terminal is higher than the threshold voltage, the memristor switches to a low resistance state (R_{on}); otherwise it switches to a high resistance state (R_{off}).[20] The MIN-MAX circuits are becoming crucial building blocks in fuzzy systems and many artificial neural networks.[21] However, the major obstacle of the existing designs is the area complexity. Thus, the memristor based MIN-MAX circuits are appearing as highly efficient circuit designs due to their nanoscale size characteristics.[22] The MIN-MAX functions are defined by the following equations:

$$\text{MIN}(V_x, V_y) = \begin{cases} V_x, & V_x \le V_y \\ V_y, & \text{otherwise} \end{cases} \quad (5)$$

$$\text{MAX}(V_x, V_y) = \begin{cases} V_x, & V_x \ge V_y \\ V_y, & \text{otherwise} \end{cases} \quad (6)$$

Assuming that V_x and V_y are the voltages at the x and y terminals respectively, Figure 4(b) shows a two Memristor Rationed Logic (MRL) MIN circuit.[20] $M1$ and $M2$ represent memristors. If $V_x > V_y$, then the voltage V follows V_y. As the current flows from the memristor $M1$ to memristor $M2$ where the voltage appearing at the positive terminal

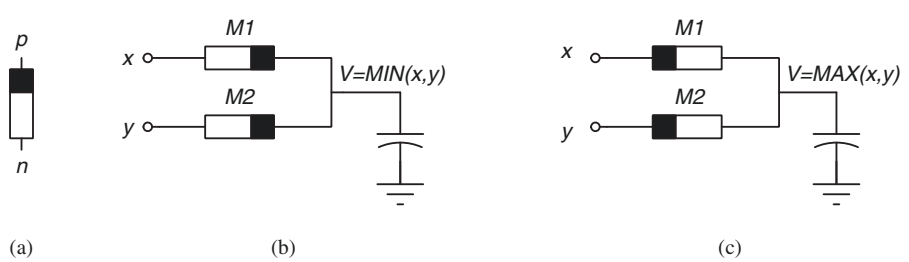

Fig. 4. (a) Single memristor. (b) Memristor based MIN circuit. (c) Memristor based MAX circuit.

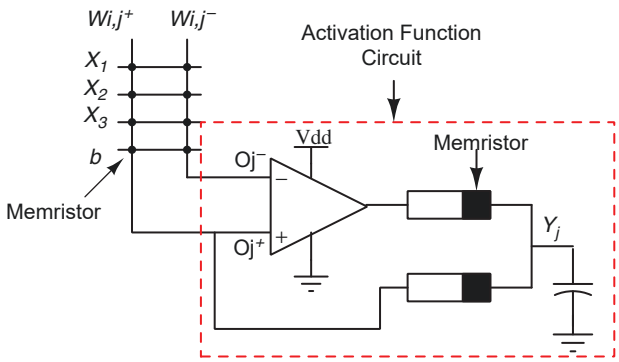

Fig. 5. A single neuron crossbar along with memristor MIN based activation function.

of $M1$ is smaller than its negative terminal $M2$. Hence, $M1$ switches to 'OFF' state and $M2$ switches to 'ON' state. If $V_x = V_y$, then the voltage V approximately equals to V_x or V_y. As $M1$ and $M2$ appear in parallel and no current flows between the inputs. If $V_x < V_y$, then the voltage V follows V_x. $M1$ switches to the 'ON' state and $M2$ switches to the 'OFF' state. Figure 4(c) shows memristor based MAX circuit. MAX circuit operations are vice-versa to the MIN circuit operations as explained above.

3. MEMRISTIVE NEURAL NETWORK CIRCUIT DESIGN

3.1. Memristive Neuron Schematic

Figure 5 shows the proposed architecture of a memristive neuron, comprising a comparator and an MRL MIN circuit, for realizing the ReLU activation function. The following equation represents the functionality of this circuit:

$$Y_j = \begin{cases} O_j^+, & O_j^+ \geq O_j^- \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

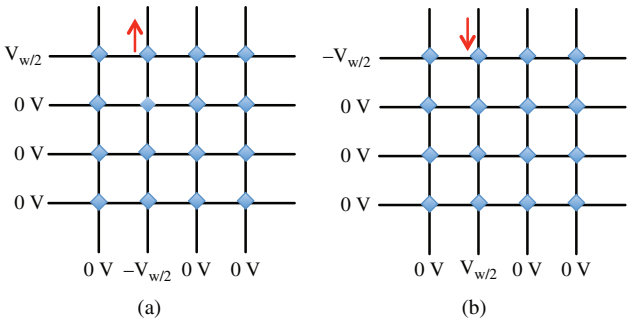

Fig. 7. Writing a single memristor in crossbar: (a) Increasing the memristor's conductance (b) decreasing the memristor's conductance.

$$W_{i,j} = W_{i,j^+} - W_{i,j^-} \tag{8}$$

where Y_j is the final output of this circuit, O_j^+ and O_j^- represent the dot product at each column representing a single neuron, as two memristors are required to represent a single weight ($W_{i,j}$). If $O_j^+ > O_j^-$, then it will give a positive effect on the total dot product (O_j). Otherwise, it will produce a negative effect. According to Eq. (7), if the total dot product produces a negative effect, then it will be considered as a 0 at the final output. Otherwise, the higher dot product value between the two columns is considered as the final output of a neuron. The total dot product of each neuron can be calculated by using the following equation:

$$O_j = O_j^+ - O_j^- \tag{9}$$

The comparator in Figure 5 provides an output of either zero or V_{dd}, where V_{dd} is assumed to be the supply voltage. Then, the memristor based MIN function is used to calculate the minimum value between the comparator output and the dot product at the column on the positive side (O_j^+).

The SPICE simulation result of the proposed activation circuit is shown in Figure 6. Here, $V(x)$ and $V(y)$ are

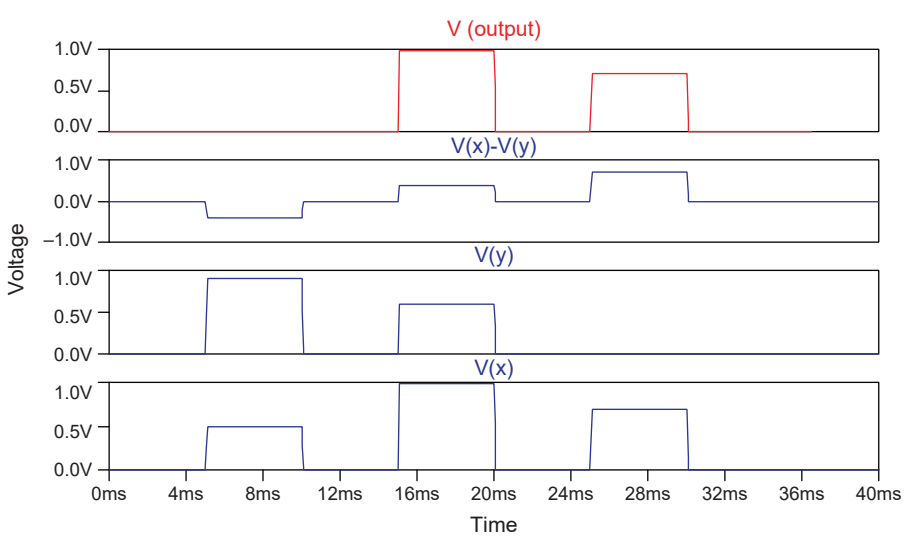

Fig. 6. Simulation results of the circuit in Figure 5.

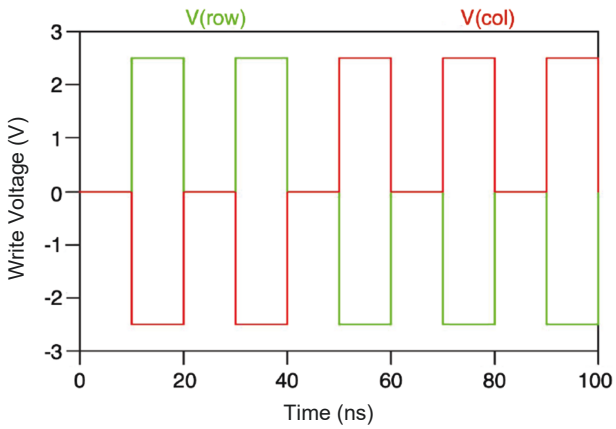

Fig. 8. Write voltage pulses.

assumed as positive (O_j^+) and negative (O_j^-) inputs of the comparator respectively. Additionally, $V(x) - V(y)$ represents the total dot product (O_j) and V(output) shows the final output (Y_j) of the circuit. If the total dot product gives a negative effect, then the final output will be zero; otherwise it follows $V(x)$.

3.2. Writing and Reading a Memristor in Crossbar Array

Programming a memristor in the crossbar includes a suitable write voltage across it. The write voltage V_w should be greater than the threshold voltage (V_{th}, considered as 4 V[23]). During a write operation, only the desired memristor should receive the V_w pulse. The write voltage ($V_{w/2}$) is applied to the selected row in which a memristor has to be written while the voltage ($-V_{w/2}$) is applied to the selected column and vise-versa in order to increase or decrease the value of the desired memristor as shown in Figure 7. The Rest of the rows and columns are set to 0 volt.

The voltage shown in Figure 8 are the write voltage pulses that are applied to the memristor crossbar. It shows two write pulses, one is applied to the row and the other is applied to the column of the desired memristor. Thus, the desired memristor achieves the voltage above threshold. The write voltage is set to 2.5 V and −2.5 V for row and column according to increase or decrease the memristance during write operation.

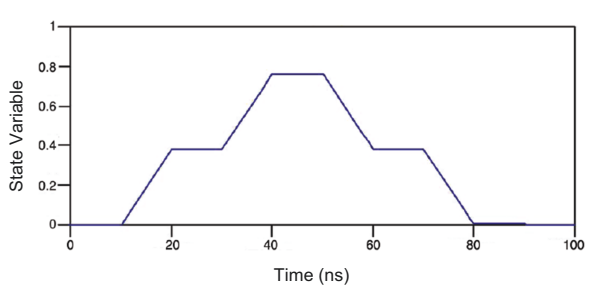

Fig. 9. Memristor state variable value.

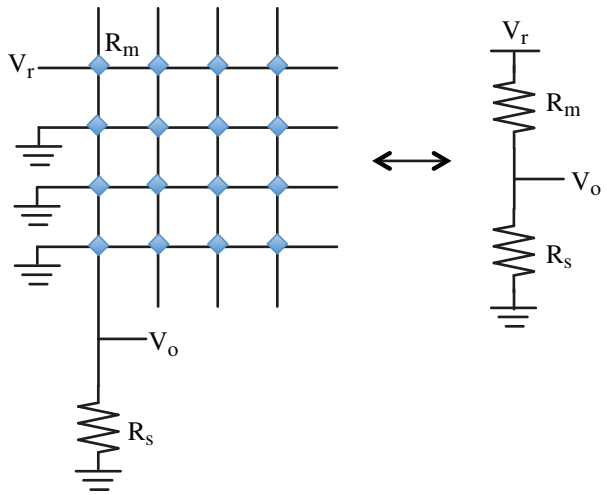

Fig. 10. Reading the desired memristor in the crossbar.

The state variable value of memristor lies between 0 and 1 and it corresponds to the change in conductance value of each memristor. The Figure 9 shows the change in state variable as a result of the applied voltages.

The read operation is a bit challenging compared to the write operation. The read voltage (V_r) should be less than the memristor threshold voltage (V_{th}) so that it does not change the state variable and hence affect its resistance. The state of a memristor can be read by using the voltage divider sensing technique. A sense resistor (R_s) is connected in series to the memristor in order to convert the current from the memristor into a voltage signal. The read voltage (V_r) is applied to the row of the desired memristor and 0 V to the rest of the rows in order to read a single memristor as shown in Figure 10. The output voltage (V_o) and current (I_o) is computed according to Eqs. (10) and (11). It represents the output of the voltage divider between sense resistor and the resistance (R_m) of the memristor itself.

$$V_o = V_r \frac{R_s}{R_s + R_m} \tag{10}$$

$$I_o = \frac{V_r}{R_s + R_m} \tag{11}$$

4. MEMRISTOR BASED MULTI-LAYER NEURAL NETWORK DESIGN

This section illustrates memristive crossbar architectures for multi-layer neural networks.

4.1. Two-Layer Circuit Design

The two-layer circuit design is utilized to implement non-separable functions as follows:

Three Bit Parity Function. A three bit parity function has been simulated using the schematic as shown in Figure 11. Here, three inputs and one bias are used. Two memristors

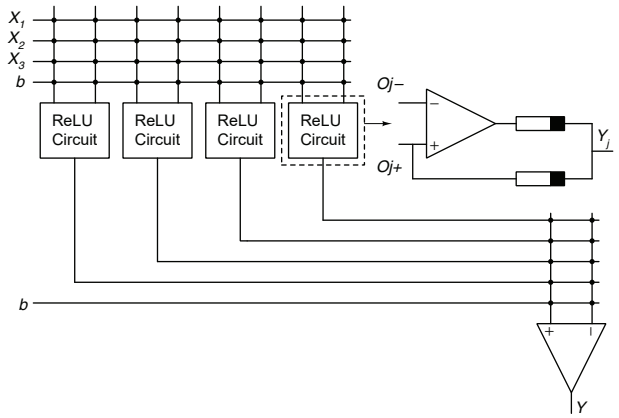

Fig. 11. Memristor based crossbar for 3-bit parity.

are used to compute a single weight value as it determines the positive and negative effect on the total dot product of neurons at each layer. The outputs of the neurons at the final layer are thresholded. Thus, a comparator is used at the output layer. In Figure 11, 4×8 memristors are used to implement crossbar at the first layer and the final layer contains 5×2 memristors along with bias. 4×2 memristors are used at hidden layer for the activation circuit. Figure 19 shows the epochs used by each activation function during training to learn a three bit parity function.

Full Adder Function. The memristor based crossbar structure for full adder function is displayed in Figure 12. The network consisting with 3-input neurons, 4-hidden neurons and 2-output neurons where the two outputs represent sum and carry respectively. 4×8 memristors are used by the first layer and 5×4 memristors are used by the output layer in the memristor crossbars. 4×2 memristors are used to implement activation function at hidden layer. Figure 20 shows the error rate of full adder function during training. The results in Figure 20 show the total number of epochs required for training using each activation function and the proposed activation function takes less time for training than other activation functions.

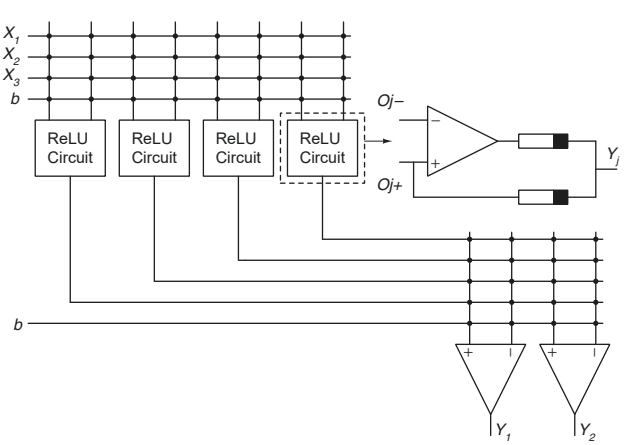

Fig. 12. Memristor crossbar structure for full adder function.

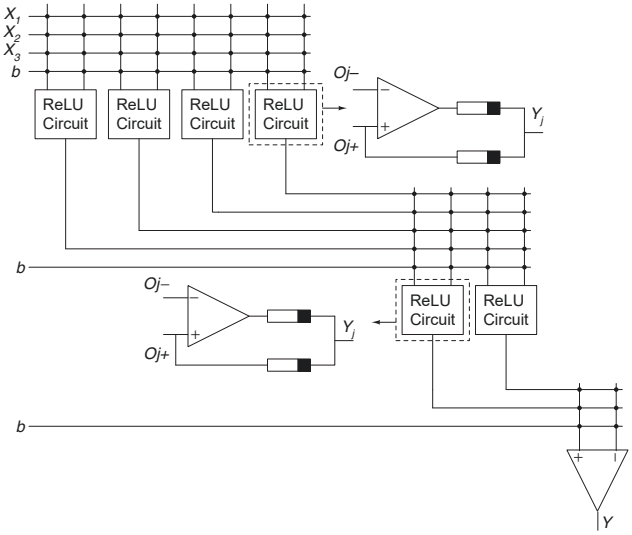

Fig. 13. Memristive crossbar circuit for parity function.

4.2. Three-Layer Circuit Design

This section explains three-layer memristor crossbar based neural network implementation. Figure 13 unveils the memristor crossbar schematic for three bit parity function. The network topology used for this function is $3 \rightarrow 4 \rightarrow 2$. The first layer consisting with 4×8 memristors; the second layer contains 5×4 memristors and the output layer contains 3×2 memristors. To implement activation circuit, two memristors are used at each layer for a single output neuron. Thus, 4×2 memristors are used at hidden-layer 1 and 2×2 are utilized at hidden-layer 2. Figure 21 shows the total epochs required for training using each activation functions.

4.3. Pattern Classifiers

The Figure 14 shows 4×4 binary image and its corresponding neural structure. The network includes 17 inputs including one bias, and two outputs. The hidden layer contains six neurons. The bias value is fixed and considered

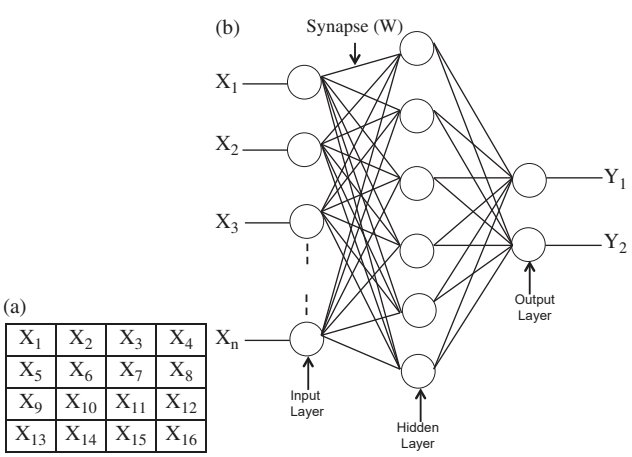

Fig. 14. (a) Binary image. (b) Multi-layer neural network structure.

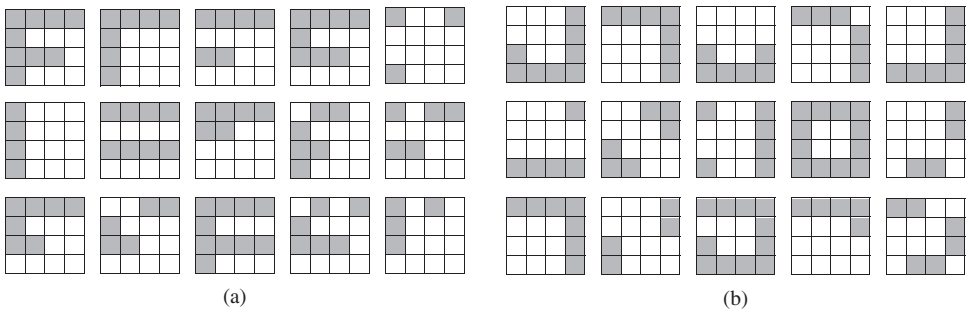

Fig. 15. (a) Set of patterns for 'F'. (b) Set of patterns for 'J'.

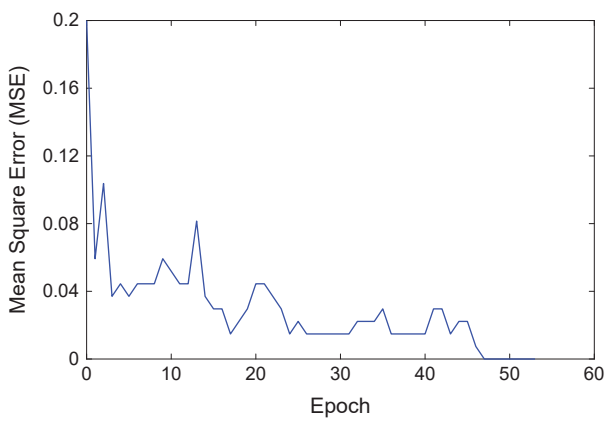

Fig. 16. The mean square error during training of iris pattern recognition.

as 1. As two memristors are used per synapse, the memristor crossbar circuit contains 17×12 memristors at hidden layer and 7×4 at the output layer.

The Figure 15 displays the set of patterns used for classification. Two alphabets 'F' and 'J' are used. The training set consists of fifty patterns and the Figure 15 shows some of the patterns. The same sets of training patterns are used for testing. Figure 15(a) represents the set of a pattern for alphabet 'F' and Figure 15(b) shows the patterns for alphabet 'J'. Noisy versions of these alphabets are also considered. The First pattern in each set of patterns is an ideal pattern and the rest are noisy versions as shown in Figure 15. The grey pixels of images are represented by 1 and white are represented by 0. The output for alphabet 'F' is considered as 1 and alphabet 'J' is 0.

The result in Figure 22 illustrates the number of epochs consumed by the network to reach zero or minimum error during the training process.

4.3.1. Iris Classification

Iris dataset[24] contains 3 classes and each class has 50 instances, where each class represents different type of iris plant called setosa, versicolour and virginica. There are four attributes. The dataset contains 150 patterns in total. First, the data is normalized and then divided into training and testing sets. 90% data are considered for training and 10% is used for testing. The network consists of four input neurons, six hidden neurons, three output neurons and one bias. The configuration for this network is $4 \rightarrow 6 \rightarrow 3$. The result in Figure 16 shows the mean square error during the training of iris dataset.

4.4. Experimental Setup

Memristor based multi-layer networks are trained and simulated. These networks are designed to implement pattern classifiers and 3-input parity functions. The memristive neural circuits shown in Figures 11–14 are first trained using a C++ environment, with a focus on its crossbar architecture, based on the backpropagation algorithm.[25] This method of learning is very efficient and robust, however, its hardware implementation is difficult.

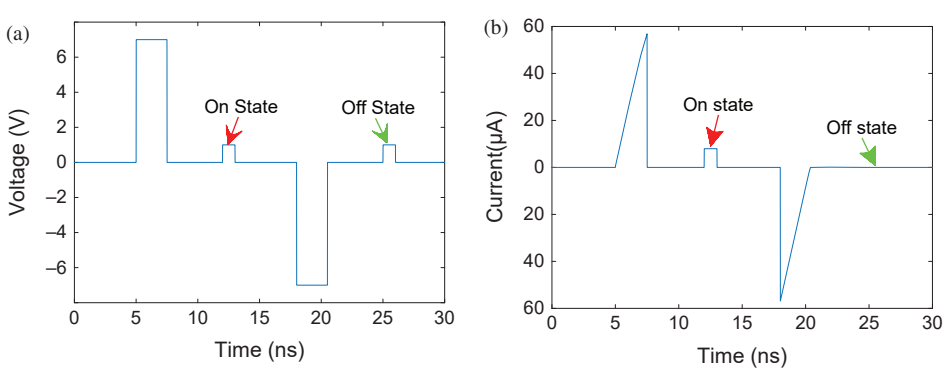

Fig. 17. Simulation results display. (a) Voltage pulse. (b) Current pulse.

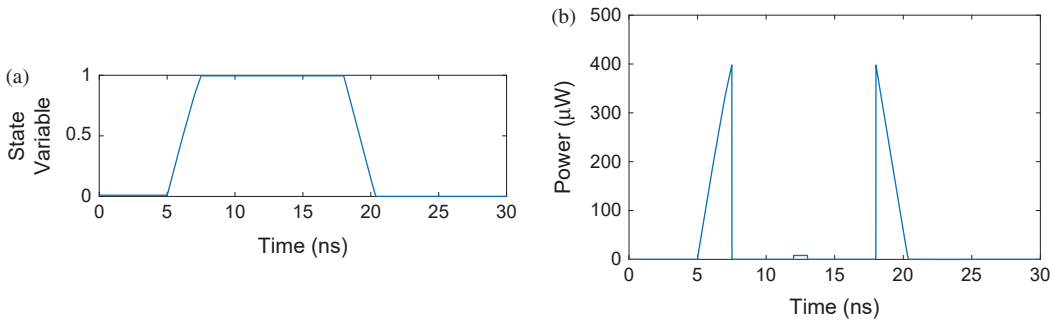

Fig. 18. Simulation results display. (a) State variable. (b) Power.

All the weights are initialized with low conductance values. After training, the final conductance values are calculated. Equation (4) is used to compute the dot product at each column as it represents the computation of a simple memristor crossbar.

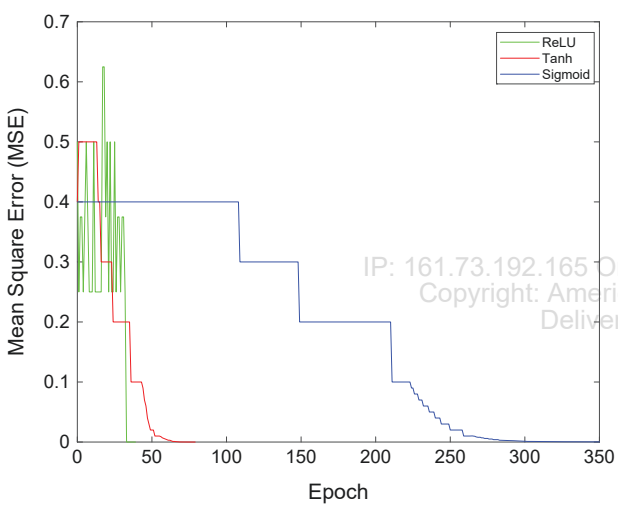

Fig. 19. Epochs required for mean square error to reach zero or minimum.

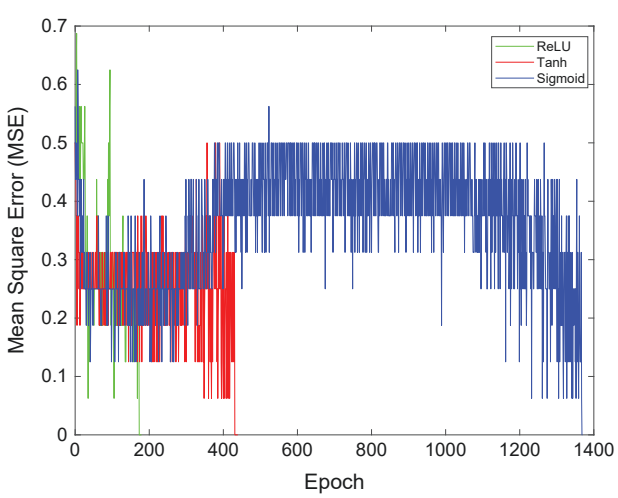

Fig. 20. Error during training process of full adder function.

Once the memristive circuit has been modeled and trained in software, the circuit is simulated in SPICE based on the memristor model in Refs. [23, 26]. This memristor device is selected as it yields higher resistance ratio ($R_{off}/R_{on} = 10^6$) as well as fast switching time, which is highly desirable for neural network implementations. The simulation results of this memrostor device are shown in Figures 17 and 18. The voltage and current waveforms are displayed in Figure 17. A +7 V/−7 V pulse is applied to switch the memristor device successfully from high resistance state into low resistance state and vice-versa. Figure 18 shows the state variable value of this device that lies between 0 and 1 and the power in this memristor device with respect to time which is calculated by multiplying the voltage and current pulses displayed in Figure 17. The state variable plot shows that the device is successfully switched by applying the voltage waveform as shown in Figure 17. These simulation results show that this device provides the low switching time in nanoseconds. Thus, it will provide more precise results while using in crossbar simulation. The R_{off} (high resistance) and R_{on} (low resistance) are 125 MΩ and 125 KΩ respectively. All the memristors in the crossbar are programmed in SPICE according to the resistance values pre-calculated in software using write and read schemes described in Section 3. The implementations in SPICE also consider the alternate

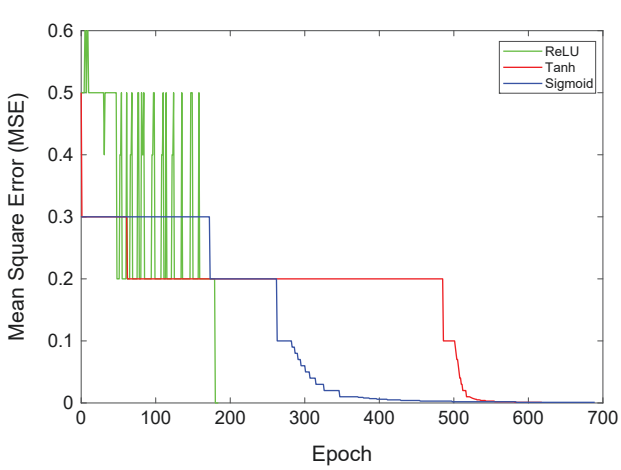

Fig. 21. The mean square error on 3-layer network during training.

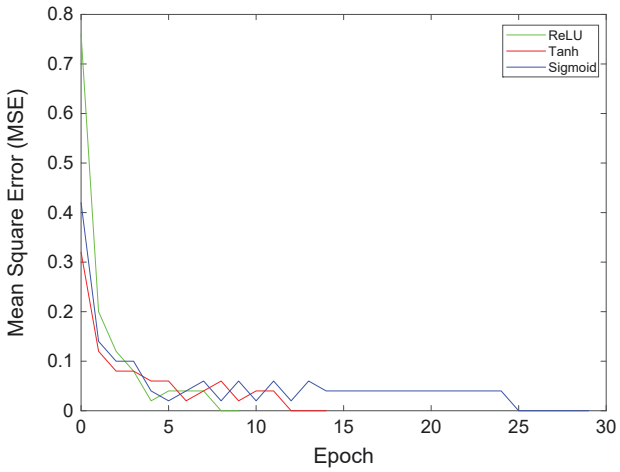

Fig. 22. The total epochs required for mean square error to reach zero.

current paths in the memristor crossbar. Thus, a sequence of alternate write and read pulses are used to program each memristor in crossbar to achieve the desired conductance value. The research[7,11] show a strong correlation between the change of memristor state and width, an amplitude of the applied voltage. The pulse width much smaller than the pulse to fully switch the memristor is used to write and the pulse of 1 V is used to read a single memristor. The 0T1M approach is used in these circuits, as it is denser and smaller in area as compared to the 1T1M circuit designs.

5. EXPERIMENTAL EVALUATIONS

The experimental results of training a three bit parity, full adder and pattern classifier are shown in Figures 19–22 respectively. The mean square error (MSE) during training reaches zero or minimum error value for ReLU much faster than the tanh and sigmoid functions. Additionally, zero error was found after crossbar simulation and testing the inputs in SPICE. A 3-bit weight precision was enough to successfully classify all the functions and patterns. Thus, the non-separable functions and pattern classifiers are successfully trained in software and tested in SPICE by using the schematics in Figures 11–14. The proposed architecture also provides significant area benefits compared to the circuits presented in Refs. [17, 18] as shown in Table I. In this table, sizes of a comparator and a memristor are generously assumed to be $750F^2$ [27] and $4F^2$ [28] respectively. As shown in columns two and three, the areas reported by the existing techniques are almost two and three times more than the proposed method just for a single neuron. Clearly, the area increases substantially as the number of neurons and hidden layers increase, e.g., as shown in the third row and fourth row of the table, which corresponds to the three bit parity function in Figures 11 and 13 respectively. Moreover, this activation function helps to reduce training time as shown in Figures 19–22. Thus, fast computations with less training time can be achieved. However, it takes more time to program the memristors in crossbar which is a generic issue in all memristor based crossbar structures.

Table I. Area (F^2) benefit of proposed technique.

	Technique of Ref. [17]	Technique of Ref. [18]	Proposed technique
Single neuron	2250	1500	758
Four neuron	9000	6000	3032
Six neuron	13500	9000	4548

6. CONCLUSION

This paper represented a novel memristive circuit for realizing the ReLU activation function. The proposed circuit comprises a single comparator and two memristors configured to realize the MIN function. Experimental results, based on a multi-layer neural network, showed that the proposed architecture requires significantly lower hardware compared to existing approaches. In addition, the results also demonstrated that non-separable functions can be trained and simulated successfully using this circuit. The proposed architecture can also reduce computational costs during the training of the network in software, as well as, help speed up the training process due to the simplicity of the underlying activation function. Therefore, the proposed approach can be much more effective for training and implementation of deep neural networks compared to the existing approaches.

References

1. L. Chua, Memristor—The missing circuit element. *IEEE Transactions on Circuit Theory* 18, 507 **(1971)**.
2. M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication, *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June **(2016)**, pp. 1–6.
3. S. P. Adhikari, H. Kim, R. K. Budhathoki, C. Yang, and L. O. Chua, A circuit-based learning architecture for multilayer neural networks with memristor bridge synapses. *IEEE Transactions on Circuits and Systems I: Regular Papers* 62, 215 **(2015)**.
4. D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, The missing memristor found. *Nature* 453, 80 **(2008)**.
5. Y. Ho, G. M. Huang, and P. Li, Nonvolatile memristor memory: Device characteristics and design implications. *ACM International Conference on Computer-Aided Design* **(2009)**, pp. 485–490.
6. G. S. Snider, Spike-timing-dependent learning in memristive nanodevices, *Proceedings of the 2008 IEEE International Symposium on Nanoscale Architectures*, IEEE Computer Society **(2008)**, pp. 85–92.
7. S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, Nanoscale memristor device as synapse in neuromorphic systems. *Nano Letters* 10, 1297 **(2010)**.
8. M. Laiho, E. Lehtonen, A. Russel, and P. Dudek, Memristive synapses are becoming a reality. *Neuromorhic Eng.* 1 **(2010)**.
9. C. Yakopcic and T. M. Taha, Energy efficient perceptron pattern recognition using segmented memristor crossbar arrays, *IEEE*

International Joint Conference on Neural Networks (IJCNN) **(2013)**, pp. 1–8.
10. M. Chu, B. Kim, S. Park, H. Hwang, M. Jeon, B. H. Lee, and B.-G. Lee, Neuromorphic hardware system for visual pattern recognition with memristor array and CMOS neuron. *IEEE Transactions on Industrial Electronics* 62, 2410 **(2015)**.
11. M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K. K. Likharev, and D. B. Strukov, Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521, 61 **(2015)**.
12. P. M. Sheridan, C. Du, and W. D. Lu, Feature extraction using memristor networks. *IEEE Transactions on Neural Networks and Learning Systems* 27, 2327 **(2016)**.
13. J. Zhou, Y. Tang, J. Wu, and X. Yi, Image segmentation with threshold based on memristors, *IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)* **(2013)**, pp. 41–44.
14. Y. V. Pershin and M. Di Ventra, Practical approach to programmable analog circuits with memristors. *IEEE Transactions on Circuits and Systems I: Regular Papers* 57, 1857 **(2010)**.
15. M. Soltiz, D. Kudithipudi, C. Merkel, G. S. Rose, and R. E. Pino, Memristor-based neural logic blocks for nonlinearly separable functions. *IEEE Transactions on Computers* 62, 1597 **(2013)**.
16. X. Glorot, A. Bordes, and Y. Bengio, Deep sparse rectifier neural networks, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, edited by G. Gordon, D. Dunson, and M. Dudk; *Proceedings of Machine Learning Research, PMLR*, Fort Lauderdale, FL, USA, April **(2011)**, Vol. 15, pp. 315–323.
17. F. Alibart, E. Zamanidoost, and D. B. Strukov, Pattern classification by memristive crossbar circuits using *ex situ* and *in situ* training. *Nature Communications* 4, 2072 **(2013)**.
18. R. Hasan, T. M. Taha, and C. Yakopcic, On-chip training of memristor crossbar based multi-layer neural networks. *Microelectron. J.* 66, 31 **(2017)**.
19. V. Nair and G. E. Hinton, Rectified linear units improve restricted boltzmann machines, *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, Omnipress, USA **(2010)**, pp. 807–814.
20. X. Yang, A. Adeyemo, A. Bala, and A. Jabir, Novel techniques for memristive multifunction logic design. *Integration, the VLSI Journal* **(2017)**. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167926016301766.
21. R. Shayanfar, A. Khoei, and K. Hadidi, Multi-input voltage-mode min-max circuit. *2010 18th Iranian Conference on Electrical Engineering* **(2010)**, pp. 413–416.
22. M. Klimo and O. Such, Memristors can implement fuzzy logic. *CoRR* **(2011)**, Vol. abs/1110.2074.
23. C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, Memristor spice model and crossbar simulation based on devices with nanosecond switching time, *The 2013 International Joint Conference on Neural Networks (IJCNN)*, August **(2013)**, pp. 1–7.
24. D. Dheeru and E. Karra Taniskidou, UCI machine learning repository. **(2017)**. [Online]. Available: http://archive.ics.uci.edu/ml.
25. S. Russell and P. Norvig, Ai a modern approach. *Learning* 2, 4 **(2005)**.
26. C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, A memristor device model. *IEEE Electron Device Letters* 32, 1436 **(2011)**.
27. C. Yakopcic, Method for *ex-situ* training in memristor-based neuromorphic circuit using robust weight programming method. *Electron. Lett.* 51, 899 **(2015)**.
28. S. H. Jo and W. Lu, CMOS compatible nanoscale nonvolatile resistance switching memory. *Nano Lett.* 8, 392 **(2008)**.

Anu Bala

Anu Bala *received the Masters of Engineering (ME) degree in Information Technology from Panjab University, Chandigarh, India and pursing the Ph.D. in Computing and Mathematics at Oxford Brookes University, Oxford, UK. Her current research focuses on Design and Analysis of Memristor based Neural Networks.*

Xiaohan Yang

Xiaohan Yang *received the M.Sc. (Hons.) degree in Electronics from the University of Edinburgh, Edinburgh, UK, now pursing the Ph.D. in Computing and Mathematics at Oxford Brookes University, Oxford, UK. Xiaohan's research area is Design Analysis and Tests of Reliable Memristor Based Logic Architectures.*

Adedotun Adeyemo

Adedotun Adeyemo *received the M.Sc. degree in Advanced Microelectronics Systems Engineering from University of Bristol, Bristol, UK in 2013 and Ph.D. degree from the School of Engineering, Computing and Mathematics, Oxford Brookes University, Oxford, UK, in 2018. He is currently a visiting researcher at School of Engineering, Computing and Mathematics, Oxford Brookes University while also working as a verification Engineer with Infineon Technologies, UK. His current research interests include circuit design and analysis as well as modelling of emerging technologies.*

Abusaleh Jabir

Abusaleh Jabir *received the M.Sc. (Hons.) degree in computer science from the University of Dhaka, Dhaka, Bangladesh and the Ph.D. degree in computing from the University of Oxford, Oxford, U.K., in 2001. He was with the Hardware Compilation Group, a subdivision of the renowned Programming Research Group at the University of Oxford. He served as a Lecturer with the Department of Computer Science, University of Dhaka. He was with Celoxica Ltd., Oxford, as a Senior Member of their research staff, while completing the Ph.D. degree. He is currently a University Reader with the Department of Computing and Communication Technologies, Oxford Brookes University, Oxford, where he established and leads the Advanced Reliable Computer Systems Group. He is the inventor/co-inventor of a number of patents being commercialized. His current research interests include electronic design automation, computer architectures, reliable and error-tolerant digital systems, in particular, emerging technologies, hardware tests and verification, and hardware for reliable cryptosystems. Dr. Jabir was a recipient of the IET/IEE Hartree Premium Award (Best Paper Award) with J. Saul for his paper Minimization Algorithm for Three Level Mixed AND-OR-EXOR/AND-OR-EXNOR Representation of Boolean Functions.*