A Memristor Based Binary Multiplier

Saeed Haghiri*, Ali Nemati*, Soheil Feizi*, Amirali Amirsoleimani**, Arash Ahmadi** and Majid Ahmadi**
*Department of Electrical and Computer Engineering, Razi University, Kermanshah, Iran
**Department of Electrical and Computer Engineering, University of Windsor, Windsor, Canada
haghirisaeed@chmail.ir {ali_nemati_1989, s.f31391}@yahoo.com
{amirsol, aahmadi, ahmadi}@uwindsor.ca

***Abstract*—Memristor is a two-terminal nanodevice that has recently attracted the attention of many researchers due to its simple structure, non-volatility behaviour, high-density integration, and low-power consumption. This paper presents and evaluates a novel binary multiplier composed of memristive devices and nanowire crossbar arrays. Using the proposed multiplier instead of usual digital circuits, the number of digital gates are a major challenge for implementing combinational logic in hybrid circuits can be reduced. The proposed memristor-based multiplier requires 20 memristor devices and it performs multiplication with 8 computational steps.**

***Index Terms*—Memristor, Material Implication, Binary Multiplier.**

I. Introduction

In 1971, Leon Chua proposed the recently discovered memristor [1] (memory resistor) as a fourth missing circuit element, displaying a connection between its resistance and the electrical charge through it [2]. Memristor has high attention of researchers due to its simple structure, non-volatility behaviour, high-density integration, and low-power consumption properties. These two-terminal resistive switches have attracted increasing interest as a suitable alternative transistors in the digital and analog electronic circuits [3]-[8].

The logic computing applications of memristor have been investigated in several publication. In-memory computing scheme is the main target application for these devices that will be an alternative to conventional Von-Neumann computing architecture. Several logic schemes [12][13][17] are presented for implementing logic circuits with memristors [9]-[17]. There are several logic blocks designed with memristor like full adder [9][14], linear feedback shift register [10], 4 by 2 compressor [11], counter [16].

In this paper, a memristor-based binary array multiplier has been proposed with material implication logic. The mathematical formulation and the memristor-based digital circuit of the binary multiplier have been proposed. The proposed implementation requires only 8 computational steps to perform multiplication. Also, 20 memristor devices have been utilized for the proposed binary array multiplier circuit. The pure memristive nature of the proposed circuit makes this design compatible for in-memory computing architectures.

The paper is organized as follows: The material implication logic and the basic memristor-based logic gates are presented in Section II. Then, we explain the binary logic multiplier and its formulation in section III. In this Section, the proposed memristor-based multiplier is presented. The paper concludes in Section IV.

II. Material Implication Logic (IMP)

One of the main applications of memristor is applying them to producing the basic logic gates due to its switching behaviors. In this approach, by considering a digital pulse voltage to the memristor terminal, we have a switch with ON (logic 1) and OFF (logic 0) states.

The logic function $\mathrm{P} \longrightarrow \mathrm{Q}$ (also known as P IMP Q or material implication) is considered for evaluating the logic gates and circuits [1], [18]. To show the IMP action, we consider P as a question and Q as the answer to that question. If question is wrong, any answer creates a true output (logic 1 or ON state). On the other hand, when the question is true, the output can be considered as Q state. As depicted in Fig. 1, the memristor logic is based upon a resistor R_{G} ($R_{\mathrm{ON}} < R_{\mathrm{G}} < R_{\mathrm{OFF}}$) connected to two memristors, named P and Q, acting as digital switches. The corresponding initial memristances P and Q acts as the inputs of the gate, while the output of the gate is the final memristance of Q (the result is written into the logic state Q that acts as an answer). The basic concept is to apply different negative voltages to P and Q, where V_{SET}, the applied voltage on Q, has a higher magnitude than V_{COND}, the applied magnitude on P ($|V_{\mathrm{COND}}| < |V_{\mathrm{SET}}|$). If P = 1 (low resistance or ON state), the voltage on the common terminal is approximately V_{COND} and the voltage on the memristor Q is approximately $V_{\mathrm{SET}} - V_{\mathrm{COND}}$, which is sufficiently small to maintain the logic state of Q. In the case of P = 0 and Q = 0 (high resistances), the applied voltage on Q is approximately V_{SET} and Q is switched ON (Q = 1). In the case of P = 0 and Q = 1, the logic state of Q is maintained. Based on the output values for each cases of IMP logic, this function is the equivalent to:

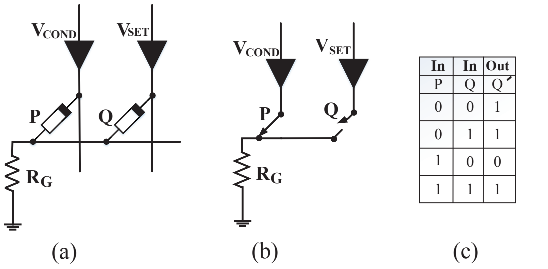
$$P \ \textbf{IMP} \ \mathrm{Q} = \bar{\mathrm{P}} + \mathrm{Q} \tag{1}$$

Consequently, the basic logic gates are presented, based on the material implication logic. In fact, the memristive implication logic (IMP) is a functionally complete operation for implementing the logic gates, NOT, AND, OR, NAND, and XOR.

A. Memristor-Based Logic Gates

In this part, the basic memristor-based logic gates are investigated for implementing the binary digital multiplier.

978-1-5090-5538-8/17/$31.00 ©2017 IEEE



In	In	Out
P	Q	Q′
0	0	1
0	1	1
1	0	0
1	1	1

Fig. 1. Illustration of the IMP operation. (a) IMP logic gate with memristors, P and Q. (b) Conditional switching circuit with memristive switches, P and Q. (c) Truth table for the IMP operation (P IMP Q).

1) **NOT** *Gate:* A digital NOT gate logical formulation can be given by

$$\mathbf{NOT}\ \ \mathrm{P} = \bar{\mathrm{P}} \tag{2}$$

The material implication (IMP) logic for implementing this gate in memristor-based form can be obtained as follows:

$$\mathbf{NOT}\ \ \mathrm{P} = \mathrm{P}\ \ \mathbf{IMP}\ \ 0 \tag{3}$$

Thus, this NOT gate can be implemented by applying two memristor devices, P and Q. In this case, at first, the memristor Q will be cleared. The sequences of output producing are illustrated in Fig. 2(a).

2) **OR** *Gate:* A digital OR gate logical formulation can be given by

$$P\ \ \mathbf{OR}\ \ \mathrm{Q} = \mathrm{P} + \mathrm{Q} \tag{4}$$

The material implication (IMP) logic for implementing this gate in memristor-based form can be obtained as follows:

$$P\ \ \mathbf{OR}\ \ \mathrm{Q} = (\mathrm{P}\ \ \mathbf{IMP}\ \ 0)\ \ \mathbf{IMP}\ \ \mathrm{Q} \tag{5}$$

Thus, this OR gate can be implemented by applying three memristor devices, P, Q, and R. In this case, at first, the memristor R will be cleared. The sequences of output producing are illustrated in Fig. 2(b).

3) **NAND** *Gate:* A digital NAND gate logical formulation can be given by

$$P\ \ \mathbf{NAND}\ \ \mathrm{Q} = \mathrm{P}\bar{.}\mathrm{Q} \tag{6}$$

The material implication (IMP) logic for implementing this gate in memristor-based form can be obtained as follows:

$$P\ \ \mathbf{NAND}\ \ \mathrm{Q} = \mathrm{P}\ \ \mathbf{IMP}\ \ (\mathrm{Q}\ \ \mathbf{IMP}\ \ 0) \tag{7}$$

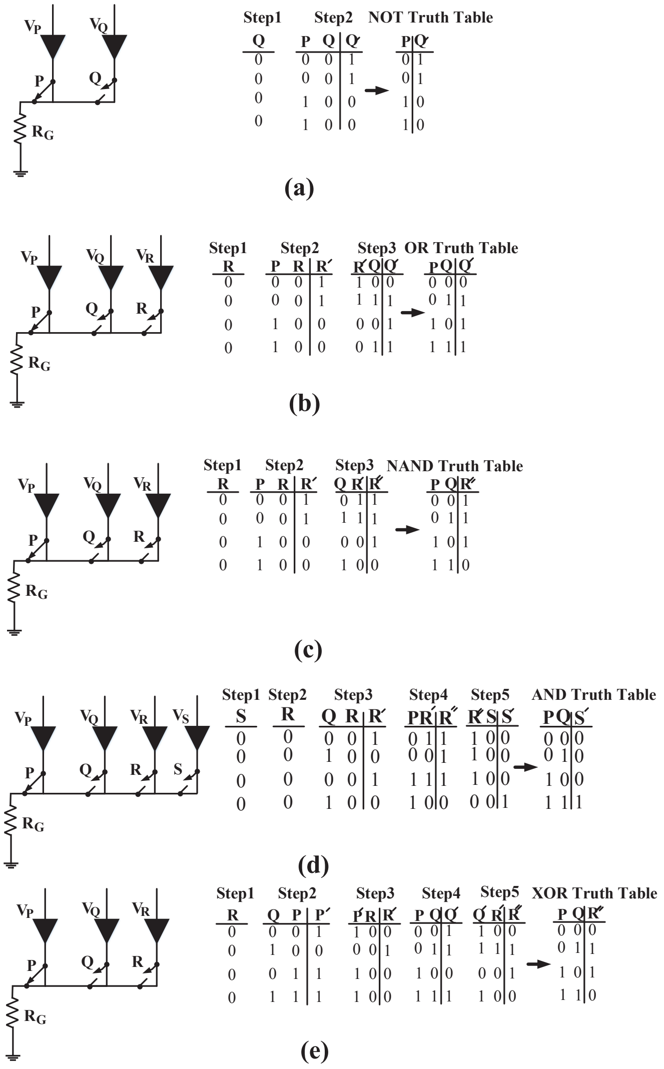
Thus, this NAND gate can be implemented by applying three memristor devices, P, Q, and R. In this case, at first, the memristor R will be cleared. The sequences of output producing are illustrated in Fig. 2(c).

4) **AND** *Gate:* A digital AND gate logical formulation can be given by

$$P\ \ \mathbf{AND}\ \ \mathrm{Q} = \mathrm{P.Q} \tag{8}$$

The material implication (IMP) logic for implementing this gate in memristor-based form can be obtained as follows:

$$P\ \ \mathbf{AND}\ \ \mathrm{Q} = (\mathrm{P}\ \ \mathbf{IMP}\ \ (\mathrm{Q}\ \ \mathbf{IMP}\ \ 0))\ \ \mathbf{IMP}\ \ 0 \tag{9}$$



Step1	Step2			NOT Truth Table	
Q	P	Q	Q′	P	Q′
0	0	0	1	0	1
0	0	0	1	0	1
0	1	0	0	1	0
0	1	0	0	1	0

Step1	Step2			Step3			OR Truth Table		
R	P	R	R′	R′	Q	Q′	P	Q	Q′
0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	1	1	0	1	1
0	1	0	0	0	0	1	1	0	1
0	1	0	0	0	1	1	1	1	1

Step1	Step2			Step3			NAND Truth Table		
R	P	R	R′	Q	R′	R″	P	Q	R″
0	0	0	1	0	1	1	0	0	1
0	0	0	1	1	1	1	0	1	1
0	1	0	0	0	0	1	1	0	1
0	1	0	0	1	0	0	1	1	0

Step1	Step2	Step3			Step4			Step5			AND Truth Table		
S	R	Q	R	R′	P	R′	R″	R″	S	S′	P	Q	S′
0	0	0	0	1	0	1	1	1	0	0	0	0	0
0	0	1	0	0	0	0	1	1	0	0	0	1	0
0	0	0	0	1	1	1	1	1	0	0	1	0	0
0	0	1	0	0	1	0	0	0	0	1	1	1	1

Step1	Step2			Step3			Step4			Step5			XOR Truth Table		
R	Q	P	P′	P′	R	R′	P	Q	Q′	Q′	R′	R″	P	Q	R″
0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0
0	1	0	0	0	0	1	0	1	1	1	1	1	0	1	1
0	0	1	1	1	0	0	1	0	0	0	0	1	1	0	1
0	1	1	1	1	0	0	1	1	1	1	0	0	1	1	0

Fig. 2. The basic memristor-based logical gates. (a) Memristor-based NOT gate. (b) Memristor-based OR gate. (c) Memristor-based NAND gate. (d) Memristor-based AND gate. (e) Memristor-based XOR gate.

Thus, this AND gate can be implemented by applying four memristor devices, P, Q, R, and S. In this case, at first, the memristors R and S will be cleared. The sequences of output producing are illustrated in Fig. 2(d).

5) **XOR** *Gate:* A digital XOR gate logical formulation can be given by

$$P\ \ \mathbf{XOR}\ \ \mathrm{Q} = \bar{\mathrm{P}}\mathrm{Q} + \mathrm{P}\bar{\mathrm{Q}} \tag{10}$$

The material implication (IMP) logic for implementing this gate in memristor-based form can be obtained as follows:

$$\left\{ \begin{array}{l} P\ \ \mathbf{XOR}\ \ \mathrm{Q} = (\mathrm{P}\ \ \mathbf{IMP}\ \ \mathrm{Q})\ \ \mathbf{IMP} \\ \quad ((Q\ \ \mathbf{IMP}\ \ \mathrm{P})\ \ \mathbf{IMP}\ \ 0) \end{array} \right. \tag{11}$$

Thus, this XOR gate can be implemented by applying three memristor devices, P, Q, and R. In this case, at first, the memristor R will be cleared. The sequences of output producing are illustrated in Fig. 2(e).

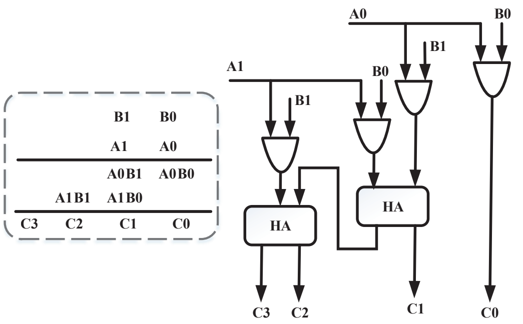


Fig. 3. Binary digital multiplier with two inputs, A and B and four outputs, C0, C1, C2, and C3.

III. Binary Digital Memristor-Based Multiplier

Fig. 3, depicts a binary array digital multiplier. This digital block consists of several sub-blocks: AND gates and Half-Adders (HA). In this case, two sub-multiplications can be collected by two HA circuits. This part presents a novel memristor-based multiplier based on the IMP and FALSE operations, which enhances the performance in the implementation of more complex logic circuits. As depicted in Fig. 3, there are four output states in binary multiplier, C0, C1, C2, and C3. Consequently, the IMP logic for these four states can be given by

A. The state C0

The material implication (IMP) logic for implementing this state in memristor-based form can be obtained as follows:

$$C_0 = A_0B_0 = (A_0 \ \mathbf{IMP} \ (\mathrm{B}_0 \ \mathbf{IMP} \ \mathrm{X}_1)) \ \mathbf{IMP} \ \mathrm{X}_2 \tag{12}$$

In this state, the memristors X1 and X2 will be cleared. Fig. 4(a), depicts the state of C0 that can be stored in memristor X2.

B. The state C1

The material implication (IMP) logic for implementing this state in memristor-based form can be obtained as follows:

$$C_1 = X_3 \ \mathbf{XOR} \ \mathrm{X}_4 \tag{13}$$

Where

$$\begin{cases} X_3 = A_0 \ \mathbf{AND} \ \mathrm{B}_1 \\ X_4 = A_1 \ \mathbf{AND} \ \mathrm{B}_0 \end{cases} \tag{14}$$

The states of X3 and X4 can be formulated in memristor form as follows:

$$\begin{cases} X_3 = (A_0 \ \mathbf{IMP} \ (\mathrm{B}_1 \ \mathbf{IMP} \ \mathrm{X}_5)) \ \mathbf{IMP} \ \mathrm{X}_6 \\ X_4 = (A_1 \ \mathbf{IMP} \ (\mathrm{B}_0 \ \mathbf{IMP} \ \mathrm{X}_7)) \ \mathbf{IMP} \ \mathrm{X}_8 \end{cases} \tag{15}$$

Thus, the memristor-based formulation can be given by

$$C_1 = (X_3 \ \mathbf{IMP} \ \mathrm{X}_4) \ \mathbf{IMP} \ ((\mathrm{X}_4 \ \mathbf{IMP} \ \mathrm{X}_3) \ \mathbf{IMP} \ \mathrm{X}_9) \tag{16}$$

In this state, the memristors X5, X6, X7, X8, and X9 will be cleared. Fig. 4(b), depicts the state of C1 that can be stored in memristor X9.

C. The state C2

The material implication (IMP) logic for implementing this state in memristor-based form can be obtained as follows:

$$C_2 = Y_1 \ \mathbf{XOR} \ \mathrm{Y}_2 \tag{17}$$

Where

$$\begin{cases} Y_1 = X_3 \ \mathbf{AND} \ \mathrm{X}_4 \\ Y_2 = A_1 \ \mathbf{AND} \ \mathrm{B}_1 \end{cases} \tag{18}$$

The states of Y1 and Y2 can be formulated in memristor form as follows:

$$\begin{cases} Y_1 = (X_3 \ \mathbf{IMP} \ (\mathrm{X}_4 \ \mathbf{IMP} \ \mathrm{Y}_3)) \ \mathbf{IMP} \ \mathrm{Y}_4 \\ Y_2 = (A_1 \ \mathbf{IMP} \ (\mathrm{B}_1 \ \mathbf{IMP} \ \mathrm{Y}_5)) \ \mathbf{IMP} \ \mathrm{Y}_6 \end{cases} \tag{19}$$

Thus, the memristor-based formulation can be given by

$$C_2 = (Y_1 \ \mathbf{IMP} \ \mathrm{Y}_2) \ \mathbf{IMP} \ ((\mathrm{Y}_2 \ \mathbf{IMP} \ \mathrm{Y}_1) \ \mathbf{IMP} \ \mathrm{Y}_7) \tag{20}$$

In this state, the memristors Y3, Y4, Y5, Y6, and Y7 will be cleared. Fig. 4(c), depicts the state of C2 that can be stored in memristor Y7.

D. The state C3

The material implication (IMP) logic for implementing this state in memristor-based form can be obtained as follows:

$$\begin{cases} C_3 = (A_0B_1A_1B_0) \ \mathbf{AND} \ (\mathrm{A}_1\mathrm{B}_1) = \\ (A_0B_1) \ \mathbf{AND} \ (\mathrm{A}_1\mathrm{B}_0) = \mathrm{X}_3 \ \mathbf{AND} \ \mathrm{X}_4 \end{cases} \tag{21}$$

This equation can be simplified as follows:

$$C_3 = Y_1 \tag{22}$$

Where

$$Y_1 = X_3 \ \mathbf{AND} \ \mathrm{X}_4 \tag{23}$$

In this state, Y3 and Y4 will be cleared. Fig. 4(d), depicts the state of C3 that can be stored in memristor Y1. Consequently, by applying the 20 memristor devices, A0, B0, A1, B1, X1, X2, X3, X4, X5, X6, X7, X8, X9, Y1, Y2, Y3, Y4, Y5, Y6, Y7, the corresponding outputs, C0, C1, C2, and C3 have been calculated in 8 computational steps.

IV. Conclusion

In this paper, a digital implementation of a memristor-based binary multiplier is presented to show how physical properties of this device can be utilized for digital circuit applications. As mentioned, memristor is a two terminal device and we require new approaches to apply it to operate as a switch in digital circuits. In this approach, material implication logic (IMP) with memristor is investigated as a useful method to create digital gates. We explained that by using this method and basic logical gates, it is possible to produce all digital functions such as binary multiplier. In this investigation, the proposed binary multiplier has been presented by employing 20 memristor devices.

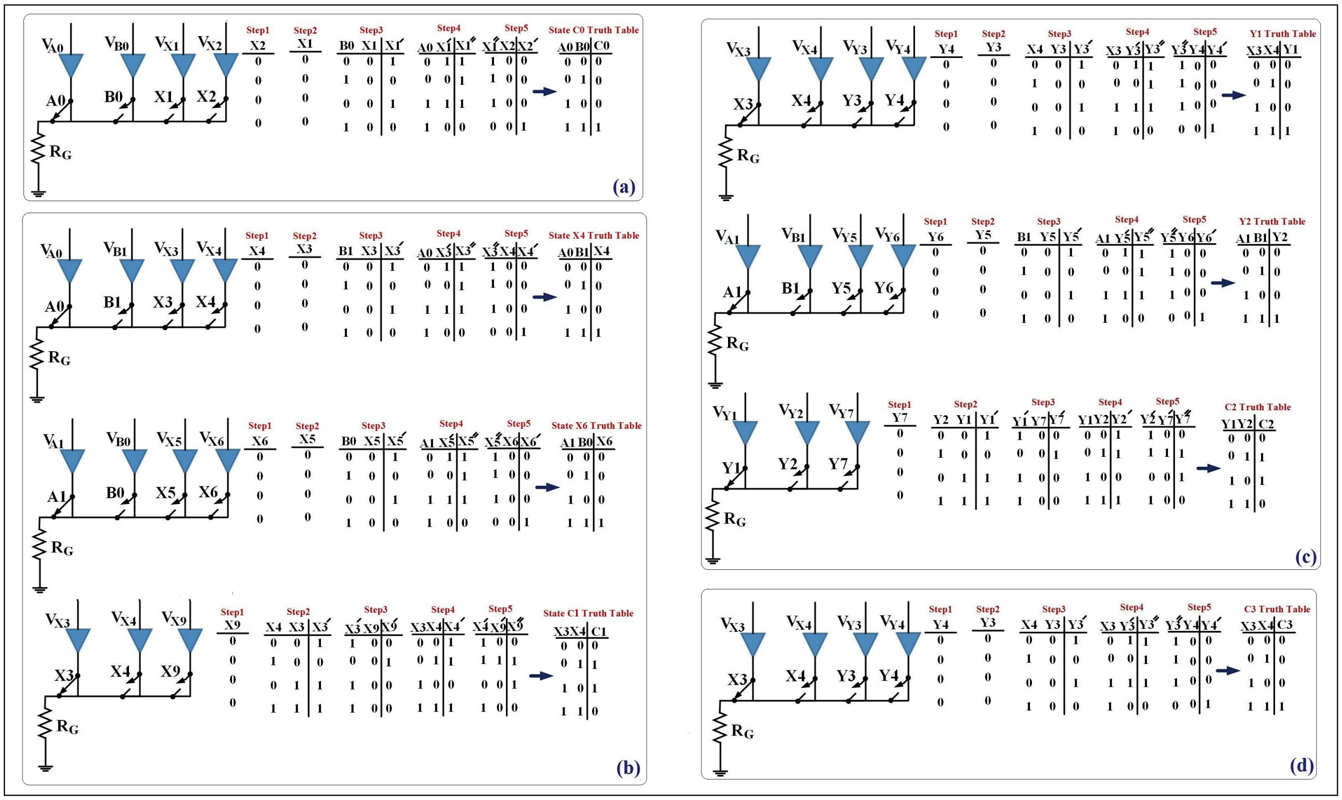


Fig. 4. Different outputs of the proposed memristor-based binary multiplier. (a) Digital circuit and sequences of the state C0. (b) Digital circuit and sequences of the state C1. (c) Digital circuit and sequences of the state C2. (d) Digital circuit and sequences of the state C3.

References

[1] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *Nature*, vol. 453, pp. 80-83, 2008.

[2] L. Chua, “Memristor-the missing circuit element,” *Circuits Theory, IEEE Transactions on [legacy, pre - 1988]*, vol. 18, no. 5, pp. 507-519, 1971.

[3] Yenpo Ho, Garng M. Huang, and Peng Li, “Nonvolatile memristor memory: device characteristics and design implications,” *In Proceedings of the 2009 International Conference on Computer-Aided Design, ICCAD 09*, pages 485-490, New York, NY, USA, 2009. ACM.

[4] M. Di Ventra, Y. V. Pershin, and L. O. Chua, “Circuit elements with memory: memristors, memcapacitors and meminductors,” *arXiv:0901.3682v1*, 2009.

[5] R. Waser and M. Aono, “Nanoionics-based resistive switching memories,” *Nature Mater*, vol. 6, pp. 833-840, 2007.

[6] A. Amirsoleimani, M. Ahmadi and M. Ahmadi, “Brain-inspired pattern classification with memristive neural network using the Hodgkin-Huxley neuron,” *Electronics, Circuits and Systems (ICECS)*, 2016 IEEE Conference on, pp. 81-84, 2016.

[7] J. Shamsi, A. Amirsoleimani, S. Mirzakuchaki, M. Ahmadi, “Modular Neuron Comprises of Memristor-based Synapse”, *Neural Computing and Applications*, pp. 1-11, 2015.

[8] S. H. Jo and W. Lu, “CMOS Compatible Nanoscale Nonvolatile Resistance Switching Memory,” *Nano Lett*, vol. 8, pp. 392-397, 2008.

[9] M. Teimoory, A. Amirsoleimani, J. Shamsi, A. Ahmadi, S. Alirezaei and M. Ahmadi, “Optimized implementation of memristor-based full adder by material implication logic,” *Electronics, Circuits and Systems (ICECS), 2014 21st IEEE International Conference on*, pp. 562-565, 2014.

[10] M. Teimoory, A. Amirsoleimani, A. Ahmadi, S. Alirezaei, S. Salimpour and M. Ahmadi, “Memristor-based linear feedback shift register based on material implication logic,” *Circuit Theory and Design (ECCTD), 2015 European Conference on*, pp. 1-4, 2015.

[11] A. Amirsoleimani, M. Ahmadi, M. Teimoory, A. Ahmadi, “Memristor-based 4:2 compressor cells design,” *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp.1242-1245, 2016.

[12] L. Guckert and E. E. Swartzlander, “MAD GatesMemristor Logic Design Using Driver Circuitry,” *in IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 2, pp. 171-175, Feb. 2017.

[13] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E.G. Friedman, A. Kolodny and U. C. Weiser, “MAGICmemristor aided LoGIC. *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11 pp. 895-899, 2014.

[14] L. Guckert and E. E. Swartzlander, “Optimized Memristor-Based Multipliers,” *in IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 2, pp. 373-385, Feb. 2017.

[15] P. L. Thangkhiew, R. Gharpinde, P. V. Chowdhary, K. Datta and I. Sengupta, “Area efficient implementation of ripple carry adder using memristor crossbar arrays,” *2016 11th International Design & Test Symposium (IDT)*, pp. 142-147, 2016.

[16] A. Chakraborty, A. Dhara and H. Rahaman, “Design of memristor-based up-down counter using material implication logic,” *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 269-274, 2016.

[17] E. Lehtonen, J. Poikonen, and M. Laiho, “Implication logic synthesis methods for memristors,” *In Circuits and Systems (ISCAS), IEEE International Symposium on*, pages 2441-2444, May 2012.