FOREX Publication
Open Access | Rapid and quality publishing

International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 3 | Pages 733-737 | e-ISSN: 2347-470X

VLSI Implementation of Hybrid Memristor Based Logic Gates

Ritesh Samanta[1], Namburi VamsiKrishna[2], Poongundran Selvaprabhu[3], Rajeshkumar V[4] and Vetriveeran Rajamani [5*]

[1]*School of Electronics Engineering, Vellore Institute of Technology, Vellore, India, ritesh.samanta2022@vitstudent.ac.in*
[2]*School of Electronics Engineering, Vellore Institute of Technology, Vellore, India, namburi.v2022@vitstudent.ac.in*
[3]*School of Electronics Engineering, Vellore Institute of Technology, Vellore, India, poongundran.selvaprabhu@vit.ac.in*
[4]*School of Electronics Engineering, Vellore Institute of Technology, Vellore, India, rajeshkumar.v@vit.ac.in*
[5]*School of Electronics Engineering, Vellore Institute of Technology, Vellore, India, vetriveeran.r@vit.ac.in*

***Correspondence:** *vetriveeran.r@vit.ac.in*; Tel.: +91-7530055178

ABSTRACT- Practical memristors have gained attention from researchers and scientists due to their potential use in a variety of electronic circuits and devices. In our paper, a hybrid Memristor-CMOS (MeMOS) logic circuit was designed and its transient response was analyzed. This circuit, which uses a N-type metal oxide semiconductor (NMOS), and P-type metal oxide semiconductor (PMOS) transistors, Operational amplifiers (OPAMPs), resistors, capacitors and multipliers replicate memristor characteristics. To facilitate the development of real memristor circuit applications, a memristor emulator is utilized for breadboard experiments. This emulator can be connected in a variety of configurations, including serial, parallel, or a combination of both, with identical or opposite polarities. By simply changing the connection, the emulator can be switched between decremental and incremental configurations. In our paper, we implemented AND logic using MeMOS. PSpice simulation of the proposed emulator have been demonstrated for TiO2 memristor model.

General Terms: Memristor emulator, AND logic, PMOS and NMOS transistors.

Keywords: Keywords Memristor-CMOS (MeMOS) Logic, Incremental, Decremental, Incremental Configuration, memristor based boolean logic, off the shelf components.

ARTICLE INFORMATION

Author(s): Ritesh Samanta, Namburi VamsiKrishna, Poongundran Selvaprabhu, Rajeshkumar V and Vetriveeran Rajamani;

Received:24/04/2023; **Accepted**: 15/06/2023; **Published**: 20/09/2023;
E- ISSN: 2347-470X
Paper Id:IJEER240411;
Citation: 10.37391/ijeer.110314
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-11/ijeer-110314.html

Publisher's Note: FOREX Publication stays neutral with regard to jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

The memristor is a new circuit element [1] which has an excellent potential for both memory and neuromorphic applications. As a non-volatile device with a size of only a few nanometers, it is well-suited for memory applications. In addition, due to its pulse-based operation and varying memristance hence making it perfect for adjusting the synaptic weights of neuromorphic cells [3, 4].

A nanoscale titanium dioxide (TiO2) device with synapse and non-volatile characteristics was created by the Stanley Williams group at Hewlett Packard (HP) which exhibited same characteristics as postulated by Chua in 1970 [1]. The main distinguishing characteristics is its pinched hysteresis loop [1] in the current versus voltage plane under sinewave excitations [1, 6] is one characteristic shared by both memristive and memristor devices. As a result of this phenomenon, the resistance of the device depends on the previous state of the input current or voltage and can function as a link in neural networks. Despite significant interest in the development of memristors, it is not expected to be commercially viable in the future due to factors like technical and cost involved in fabricating nanoscale devices [4, 5].

As result, circuit alternatives that mimic memristors are needed to build applications that take advantage of the potential of memristors. Previous research has focused on creating spice macro models for simulating memristors, but these models are not hardware, hence they cannot be used to create memristor application circuits that can be physically implemented. To address this need, researchers have developed memristor emulators, which are circuits that behave like real memristors [2, 5-6, 9-13]. However, these emulators have limitations, including limited interoperability with other analog circuit elements, difficulty in connecting them in serial, parallel, or hybrid configurations, and unstable memristance over time. In our paper, we propose a memristor emulator [2] created using commercial solid state hardware that overcomes these limitations. Our emulator faithfully reproduces the features of TiO2 memristors, has a stable memristance over a considerable amount of time and is compatible with other circuit-based technologies [5, 8]. We conduct simulations and tests to illustrate the many features of our emulator. *Figure 1* describes the relation between three circuit elements (*i.e.*, resistor, inductor, and capacitor) with four different parameters viz., flux, charge, voltage and current, *i.e.*, resistor is related by voltage and current, inductor is related by flux and current, capacitor is related by voltage and charge but there is a missing relationship between flux and charge is actually a Memristor (represented by red color letter '*M*') [1, 3].

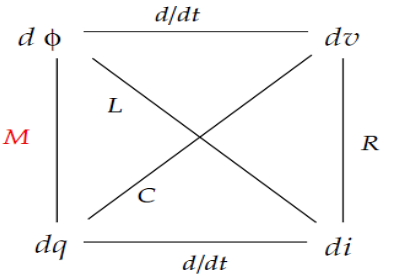

Figure 1: The fourth ideal element

2. MODELLING OF MEMRISTOR

In our designed memristor, current and voltage equation is be defined as below:

$$v(\mathrm{t}) = \mathrm{R}(\mathrm{t}).\mathrm{i}(\mathrm{t}) = \frac{d\varphi}{dq} i(\mathrm{t}) \qquad (1)$$

Here flux *φ(t)* and charge *q(t)* denoted with respect to time. As a result, the resistance may be understood as the gradient of the memristor *(φ - q)* curve at the operational point $q = q_0$ in time.

The resistance will change depending on the operating point if the curve *(φ - q)* is nonlinear. Without applied external current or voltage, the operating point remains same, thereby the resistance does not vary with time. As a result, the signal is stored in memory as the memristor's resistance or *M* value.

The memristance can be changed by changing input across the memristor since the flux f is controlled by $\varphi(\mathrm{t}) = \int_{-\infty}^{t} v(\tau) d\tau$

$$R = M = \left.\frac{d\varphi}{dq}\right|_{(q_Q\ ,\varphi_Q)} \qquad (2)$$

The TiO_2 memristor's structure is depicted in *figure 2(a) [3]*. In the TiO_2 memristor, a thin Titanium-dioxide TiO_2 layer and a thin oxygen poor Titanium-dioxide TiO_{2-x} layer are sandwiched between two platinum (pt) electrodes. The terms "undoped" and "doped" refer to the TiO_2 layer and the $TiO_{(2-x)}$ layer, respectively [1]. The boundary between the TiO_2 and $TiO_{(2-x)}$ layers varies when a signal is supplied to the device, depending on the voltage or current. The resistance between the electrodes changes as a result.

Here *D* and *w* stand for thicknesses of the sandwiched and doped (oxygen-deficient) regions, respectively, in the Titanium-dioxide memristor. Also, R_{ON} and R_{OFF} stand for the resistances (*R*) in the regions with varying dopant concentrations.

The voltage and current are related as below,

$$v(\mathrm{t}) = \left(\mathrm{R}_{\mathrm{ON}} \frac{w(\mathrm{t})}{D} + \mathrm{R}_{\mathrm{OFF}}\left(1 - \frac{w(\mathrm{t})}{D}\right)\right) i(t) \qquad (3)$$

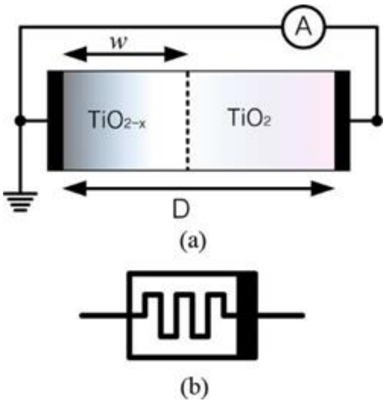

Figure 2: TiO_2 Memristor

Figure 2: (a) The Design of TiO_2 memristor is like sandwiching two layers of TiO_2 and $TiO_{(2-x)}$ in-between the two platinum(pt) electrodes. Whenever any input voltage (V), or current (I) is applied, the memristance ((Memristor resistance (M) in Ohms) /memductance (memristors conductance in siemens) is altered. (b) Memristor symbol.

Here, the state variable is $\frac{w(\mathrm{t})}{D}$. The rate at which the TiO_2 memristor's state variable changes [3] is specified as a function of current, specifically,

$$\frac{dw(\mathrm{t})}{dt} = \mu_v \frac{R_{ON}}{D} i(\mathrm{t}) \qquad (4)$$

Here, μ_v is known as dopant mobility. Since the width's velocity is linearly proportional to the current, this model is known as a linear drift model. The formula explains the relationship between the memristor's flux and charge in this case is given in *equation (5).*

$$\varphi(\mathrm{t}) = R_{OFF}\left\{q(\mathrm{t})\left[1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right] - \frac{\mu_v R_{ON}}{2D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right) q(\mathrm{t})^2\right\} + \varphi_0 \qquad (5)$$

By combining *equations (2) and (5),* we get

$$M = \frac{d\varphi}{dq}$$

$$= R_{OFF}\left\{\left[1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right] - \frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right) q(\mathrm{t})\right\}$$

$$\approx R_{OFF}\left\{1 - \frac{\mu_v R_{ON}}{D^2} q(\mathrm{t})\right\} \qquad (6)$$

We see that in *equation (6),* the memory resistance is directly proportional to the function of charge. A black thick line at one end of the memristor symbol in *figure 2(b)* designates the polarity. As a result, current moves from the left to the right (thick black line) side, as memristance *M* decreases.

2.1 Working

An input voltage is transformed into an input current when it is applied to a memristor emulator using a resistor and op-amp, which is then constrained by a virtual ground. The current is then replicated using current mirrors, which allow single-direction currents to be copied. However, for sinusoidal input currents, the circuit must be divided into positive and negative parts and fed independently in each part of the circuit. The circuit depicted in *figure 3*, the positive part of the current is copied using current mirroring circuits that is MN_0 and MN_2, and is supplied into a capacitor *C* and a resistor *R*. The negative part of the current is processed by current mirrors MP_0 and MP_2 and flows out of the resistor *R* and capacitor *C*. One of the defining characteristics of a memristor is its ability to retain the existing value for a long period till new inputs are given. The memristor emulator's programmed information is represented by the charge stored in capacitor *C*. The output terminal's wire is routed to the metal oxide semiconductor field effect transistors (MOSFET's) buffer *U1* gate to stop the discharge of the capacitor when there is no input signal.

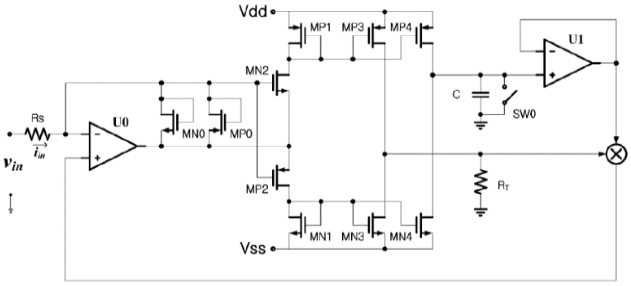

Figure 3. Hybrid Memristor configuration

2.2 Logic Implementation using Memristor

Memristor resistance is influenced by the direction of current flow, making them useful for logic computation. By using this property, voltage divider circuits can be created. This can be utilized to design AND gates using memristors. To integrate memristor with CMOS, a hybrid Memristor-CMOS (MeMOS) logic can be used, where voltages are used as the logic state. The memristors act as computational elements, rather than computational and storage elements as in material implication logic. This allows for compatibility with current generation CMOS processes and for working with the same voltage levels.

The computation of the AND operation for all input cases of a two-input AND gate using memristors only is illustrated in *figure 4*.

In *figure 4*, for the case of A=1 and B=1, both inputs are connected to V_{CC}, resulting in no current flow through the circuit and an output of *logic 1*. Similarly, for the case of A=0 and B=0, then there is no flow of current and therefore output is *logic 0*. These two cases are also applicable for the OR logic. When any one of the inputs is at *logic 1* and the other is at logic 0, current flows from VCC to GND, which modifies the resistance of the memristors and results in an output of *logic 0* for the AND gate. This calculation can be determined using the voltage divider rule.

The voltage divider circuit's output voltage at Y can be calculated as follows:

$$Y = V_{CC} \times \frac{R_{ON}}{R_{ON} + R_{OFF}}$$

R_{OFF} is considerably higher than R_{ON}. The *equation* can be made simpler as,

$$Y = V_{CC} \times \frac{R_{ON}}{R_{OFF}} \square\ V_{CC} \approx GND$$

It is also noted that this topology can be used to implement 'n' input gates using memristors, however, the primary challenge is the lack of the NOT operation, making it impossible to implement Boolean functions. One solution is to use a CMOS inverter to implement the NOT operation is illustrated in *figure 5*, however, this has trade-offs such as large area consumption, static power dissipation, delay. Alternatively, buffer circuits can be used to restore the logic level, but it is important to consider the trade-off between restoring the logic level and the associated power dissipation, delay and area consumption. It is also noted that this topology can be used to implement 'n' input gates using memristors, however, the primary challenge is the lack of the NOT operation, making it impossible to implement Boolean functions. One solution is to use a CMOS inverter to implement the NOT operation is illustrated in *figure 5*, however, this has trade-offs such as large area consumption, static power dissipation, delay.

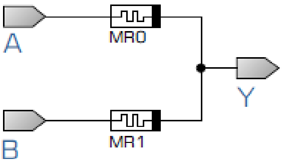

Figure 4. AND operation using memristor using voltage divider

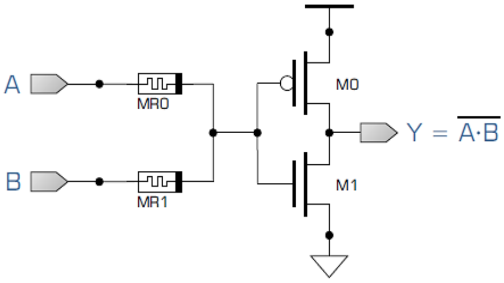

Figure 5. Memristive NAND gate structure using Hybrid Memristor-CMOS logic. The output of the CMOS NOT gate is used to obtain NAND operation, and the memristors are configured to offer NAND operation.

Alternatively, buffer circuits can be used to restore the logic level, but it is important to consider the trade-off between restoring the logic level and the associated power dissipation, delay and area consumption.

3. RESULTS

We have simulated our hybrid memristor using CMOS (MeMOS) Logic in OrCAD Cadence tool in a decremental memristor connection, the resistance of the memristor is slowly changed by applying various frequencies to a specific value of 5V applied voltage. The output of the device is a voltage or current that is dependent on the applied signal and the resistance of the memristor varies accordingly that is pinched hysteresis loop changes.

The output of our memristor circuit at different frequencies which were simulated at 10Hz (*figure 6*), 20Hz (*figure 7*), 100Hz (*figure 8*), and 1000Hz (*figure 9*), behaved like the properties of an actual memristor device using the circuit which we have implemented using hybrid CMOS (MeMOS) Logic. The properties of a memristor circuit was different at different frequencies due to the frequency-dependent properties of the memristor. The input resistance was kept at 16K, and the RC network value of R was kept at 4k and Capacitance was kept at 0.1µF.

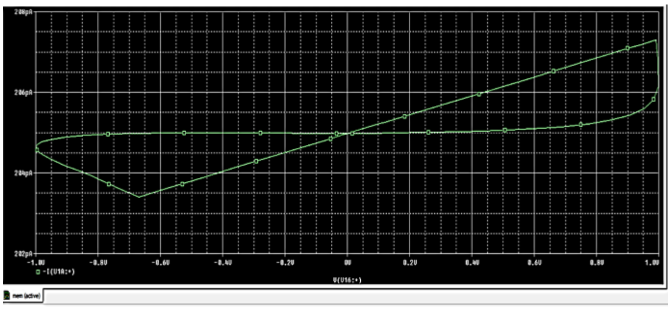

Figure 6. Memristance pinched loop at 10Hz

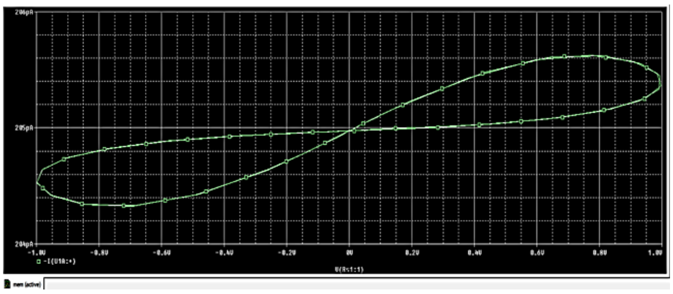

Figure 7. Memristance pinched loop at 20Hz

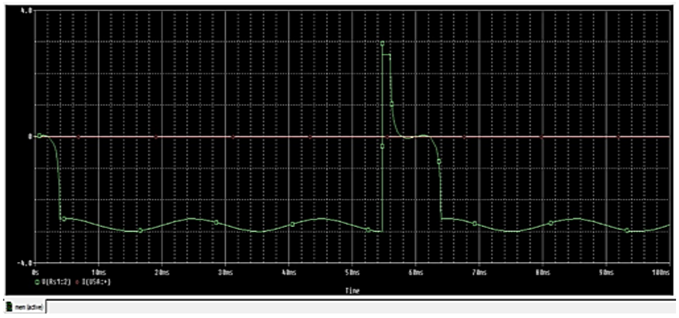

Figure 8. Memristance pinched loop at 100Hz

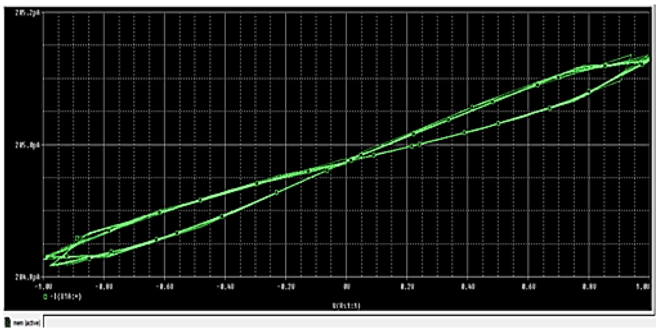

Figure 9. Memristance pinched loop at 1000Hz

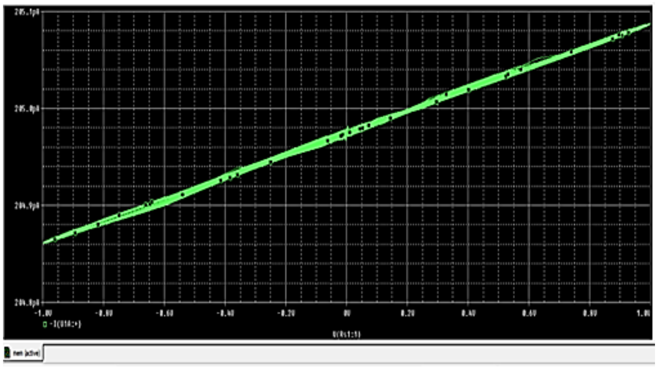

Figure 10. Memristor pinched loop for AND Logic

Using the circuit in *figure 3* and *figure 4*, we implemented an AND gate by connecting the device in series with each input signal and then measuring the output current. Note that, in Figure 4, the two decremental memristors [11] are connected in parallel. When both input signals are "1" (or "high"), the memristors will have a low resistance, allowing a high current to flow through the circuit, resulting in a high output signal. When either input signal is "0" (or "low"), the corresponding memristor will have a high resistance, blocking the current flow and resulting in a low output signal. *Figure 10 and 11* represents the output of *figure 4*.

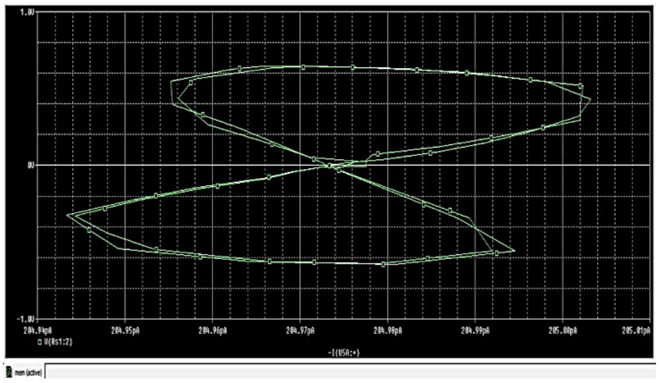

Figure 11. AND Logic waveform using transient analysis

4. DISCUSSION

Related to Memristor, numerous proposed architectures and algorithms are available till date. The architectures developed till now, uses various software to mimic a memristor but these are practically not implemented in real world. Hence, our idea was to implement memristor using Memristor-CMOS (MeMOS) Logic such that the memristor is available physically with existing available CMOS technology, for fast computing using Memristor-CMOS (MeMOS) Logic in PSpice. The results simulated are similar to that of CMOS logic with applications in terms of artificial neural networks and memory devices can be further implemented.

5. CONCLUSIONS

In this paper, a memristor circuit has been designed and simulated using readily available off-the shelf devices. This emulator can serve as a useful tool for developing memristor-based circuits. The performance of the emulator has been evaluated through circuit measurements and simulations, and has been found to closely mimic the behavior of real TiO2 memristors. Additionally, the emulator has been shown to be nonvolatile over the duration of the experiments. The emulator is also expandable, as it can be connected in various configurations, such as serial, parallel, or a combination of both, with memristors that have the same polarity as them or the opposite. By changing the connections, it is also simple to adjust the setup between incremental and decremental modes.

5. ACKNOWLEDGMENTS

Authors would like to thank Vellore Institute of Technology, Vellore, Tamilnadu, India for providing the facilities to carry out this research.

REFERENCES

[1] Chua, L. 1971. Memristor-the missing circuit element. IEEE Transactions on circuit theory, 507-519.
[2] Kim, H., Sah, M. P., Yang, C., Cho, S., and Chua, L. O. 2012. Memristor emulator for memristor circuit applications, IEEE Transactions on Circuits and Systems I: Regular Papers, 2422-2431.
[3] Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. 2008. The missing memristor found, Nature, 80-83.
[4] Snider, G. S., 2008. Self-organized computation with unreliable, memristive nanodevices. Nanotechnology, 365202-365125.
[5] Singh, T. 2015. Hybrid memristor-cmos (memos) based logic gates and adder circuits. arXiv preprint, arXiv:1506.06735, 1-10.
[6] Rajamani, V., Yang, C., Kim, H., and Chua, L. 2016. Design of a low-frequency oscillator with PTC memristor and an inductor. International Journal of Bifurcation and Chaos, 1630021-1-27.
[7] Guckert, L. and Swartzlander, E. E. 2017 Optimized memristor-based multipliers. IEEE Transactions on Circuits and Systems I: Regular Papers, 373-385.
[8] Sharma, G., and Bhargava, L. 2016. CMOS-memristor inverter circuit design and analysis using Cadence Virtuoso. IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 1-5.
[9] Biolek, Z. Biolek, D. and Biolkova, V. 2009. SPICE model of Memristor with nonlinear dopant drift. Radio Eng., 210–214.
[10] Batas, D. and Fiedler, H. 2011. A memristor SPICE implementation and a new approach for magnetic flux-controlled memristor modeling. IEEE Trans. Nanotechnol., 250–255.
[11] Kvatinsky, S., Kolodny, A., Weiser, U. C., and Friedman, E. G. 2011. Memristor-based IMPLY logic design procedure. IEEE 29th International Conference on Computer Design (ICCD), 142-147.
[12] Ghosh, M., Mondal, P., Borah, S.S., and Kumar, S. 2023. Resistor less Memristor Emulators: Floating and Grounded Using OTA and VDBA for High-Frequency Applications in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 978-986.
[13] Kumar, A., Chaturvedi, B., and Mohan, J. 2023. Minimal realizations of integrable memristor emulators. J Comput Electron, 2023 504–518.

© 2023 by the Ritesh Samanta, Namburi VamsiKrishna, Poongundran Selvaprabhu, Rajeshkumar V and Vetriveeran Rajamani. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).