Journal of Circuits, Systems, and Computers
Vol. 31, No. 3 (2022) 2250047 (19 pages)

DOI: 10.1142/S0218126622500475

World Scientific
www.worldscientific.com

A Novel Refreshment Circuit for 2T1M Neuromorphic Synapse*

Mai M. Goda[†,§], Ahmed H. Hassan[†,¶], Hassan Mostafa[†,‡,‖] and Ahmed M. Soliman[†,**]

[†]*Electronics and Communications Department, Faculty of Engineering, Cairo University, Giza, Egypt*

[‡]*University of Science and Technology, Nanotechnology and Nanoelectronics Program, Zewail City of Science and Technology, October Gardens, 6th of October, Giza 12578, Egypt*
[§]*Mai.Mostafa.j@eng-st.cu.edu.eg*
[¶]*ahhamed93@hotmail.com*
[‖]*hmostafa@uwaterloo.ca*
[**]*asoliman@idsc.net.eg*

Received 6 January 2021
Accepted 30 July 2021
Published 13 September 2021

Neuromorphic systems are the future computing systems to overcome the von Neumann's power consumption and latency wall between memory and processing units. The two main components of any neuromorphic computing system are neurons and synapses. Synapses carry the weight of the system to be multiplied by the neuromorphic attributes, which represent the features of the task to be solved. Memristor (memoryresistor) is the most suitable circuit element to act as a synapse. Its ability to store, update and do matrix multiplication in nanoscale die area makes it very useful in neuromorphic synapses. One of the most popular memristor synapse configurations is the two-transistor–one-memristor (2T1M) synapse. This configuration is very useful in neuromorphic synapses for its ability to control reading and updating the weight on a chip by signals. The main problem with this synapse is that the reading operation is destructive, which results in changing the stored weight value. In this paper, a novel refreshment circuit is proposed to restore the correct weight in case of any destructive reading operations. The circuit makes a small interrupt time during operation without disconnecting the memristor, which makes the circuit very practical. The circuit has been simulated by using hardware-calibrated CMOS TSMC 130 nm technology on Cadence Virtuoso and linear ion drift memristor Verilog-A model. The proposed circuit achieves the refreshment task accurately for several error types. It is used to refresh 2T1M synapse with any destructive reading signal shape.

Keywords: Neuromorphic computing; memristor; memristor-based synapse; refreshment circuit.

*This paper was recommended by Regional Editor Giuseppe Ferri.
‖Corresponding author.

1. Introduction

Neural networks (NNs) are considered very useful computing units in edge computing (near-sensor computing) for their high speed and small size than vonNeumann computers.[1,2] Internet of Things (IoT), for example, needs a huge effort for near-sensor computation.[3,4] This requires large area and power consumption, which make them unsuitable for small dimensions of IoT and limited power budget in sensor computing applications.[1,5] The computing unit in the NN is called neuromorphic computing. The inspiration for this name comes from the biological brain as it contains neurons. The basic two elements of the neuromorphic system are neurons and synapses. Synapse is the connected medium between two neurons which delivers neural signal from neuron to another. The neuron is responsible for taking actions depending on the neural signal from previous connected synapse.[1] This is the simplest type of NN; however, there are different types of NNs in engineering depending on the target application. Convolutional neural networks are very useful in image processing tasks.[6,7] Spiking NNs (SNNs) mimic the human neural activity in terms of time difference between neurons.[8]

Applying the NN types mentioned above with a complementary metal–oxide–semiconductor (CMOS) or field programmable gate array (FPGA) will consume large area and power.[1] This is not suitable for sensor computing applications as mentioned earlier. For those two main reasons, applying new emerging devices, such as a memristor, is essential. A memristor, or memory resistor, which has been fabricated more than ten years ago,[9] is widely used in neuromorphic synapses.[1] Its ability to combine both processing and memory in one place makes it a very promising replacement to the CMOS technology transistors. The processing complexity of most machine learning (ML) tasks is focused on matrix multiplication. The multiplication is performed between the ML input (attributes) and NN synapse's weight matrices. The memristor carries the NN weight and updates its value corresponding to the NN training algorithm.[1,10]

There are many configurations for memristor synapses;[1] the designer should select one of them based on the target system function and scale. Emerging memristor memory devices can write energy far lower than that of conventional flash memory. The small resistance ratio and wide distribution of cell resistance values in emerging memory devices results in a small difference in signals between nodes during restore operations (power on). Memristor-based synapse implementations enable energy-efficient parallel computing for matrix-vector multiplication (weight sum) using memristor arrays to accumulate a combination of input signals and synapse weightings.[11]

Emerging memristive device as an alternative to CMOS devices shows how such devices enable novel computing models that will solve the challenges of today's architectures for certain applications. It can replace the CMOS due to many advantages such as near-zero standby power, high device scalability, high integration density and CMOS process compatibility. Memristive devices provide many

opportunities not only to enable next-generation nonvolatile memories (with fast access speed, up to 1 TB storage capacity and multi-level capability) but also to enable novel alternative computing architectures required for emerging applications, such as neuromorphic systems and computation-in-memory architectures.[12]

A high-performance CMOS-memristive circuit mimics a number of essential learning properties of biological synapses. Many previous CMOS spike timing–dependent plasticity synapse circuits occupy a large silicon area, even if the synaptic weight storage is not considered.

The hybrid CMOS-memristor device area represents a factor of ten reductions in area with respect to prior CMOS art, which integrated with silicon neurons in a crossbar array structure is amenable to large-scale neuromorphic architectures.[13]

The 1 M crossbar array has been used for the first time in Ref. 14. Its size is small compared with the recent memristor synapses architecture; however, it suffers from sneak path problems and does not have negative weight capability. Another configuration is with 2 M crossbar,[15] which achieves negative weights but with double the size of 1 M. The third configuration is 2M with one resistor (2M1R), which is useful in SNNs.[16] One memristor is used for controlling, and the other one is used for storing, and the resistance is used to preserve the weight in the second memristor from different applied biases on synapses. One recent configuration is the two-transistors–one-memristor (2T1M) synapse[10] shown in Fig. 1. This configuration allows on-chip control rather than memristor-only synapses, which require additional control circuit and disconnection of memristor during control from the crossbar array.[1]

The reading signal used in Ref. 10 assumes that the square wave is symmetric (i.e., the square wave positive (ON) time equals exactly the negative (OFF) time), which is unpractical.[17] This leads to destructive reading a refreshment circuit for 2T1M synapse due to that destructive reading procedure which changes the stored weight after many cycles of operation,[18] as illustrated in Sec. 2. This paper proposes a refreshment circuit for 2T1M synapse due to that destructive reading.

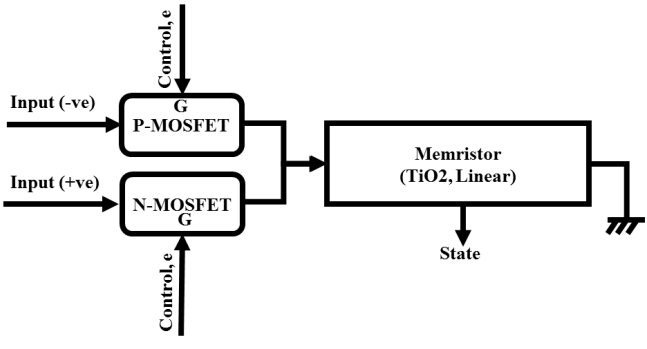

Fig. 1. The memristor-based synapse used in Ref. 10.

Taking the advantage of on-chip controlling of the synapse by the two transistors, the refreshment circuit is also on-chip (online).[1] Therefore, there is no need to interrupt the neuromorphic system by taking the synapse off the chip for refreshment. The only interrupt is the refresh writing time, which occurs on-chip, as illustrated in Sec. 3. The time of writing interrupt is insignificant compared to the original writing phase of the synapse and, correspondingly, does not affect the performance. This keeps the system performance time almost the same without the refreshment circuit and with higher accuracy. This is because the refreshment circuit captures the mismatch earlier than the time of failure in Ref. 18.

The rest of the paper is organized as follows. The 2T1M synapse in Ref. 10 and the destructive reading problem are illustrated in Sec. 2. The proposed refreshment circuit is described and analyzed in Sec. 3. Simulation results and discussions are given in Sec. 4. The process–voltage–temperature (PVT) performance of the proposed refreshment circuit is presented in Sec. 5. A single layer design of a simple 2×2 synaptic grid circuit is presented in Sec. 6. A comparison with other works is given in Sec. 7. Finally, the paper is concluded in Sec. 8.

2. 2T1M Synapse

The synapse shown in Fig. 1 consists of 2T1M.[10] The memristor model used is linear ion drift,[9] with its Verilog-A model used in the simulation results.[19] The two metal-oxide-semiconductor field-effect (MOSFET) transistors are controlling the read and write operations of the NN's weight, when the memristor is storing the weight. All the variable definitions used with the memristor are listed in Table 1.

The control signal e, as well as the input signal, guarantees that the two transistors are in the deep triode region. Therefore, the input of the memristor is almost the same as the input of the transistors (i.e., the ON resistance of the transistors is very small). The I–V characteristic of the memristor is shown in Fig. 2, which is compatible with the linear ion drift memristor model.[9] The read and write processes of the synape's weight into the memristor are controlled by the control signal e, as shown in Fig. 3(a). During the reading phase, the weight stored is being read by applying a symmetric square wave of amplitudes of $\pm 1.8\,\text{V}$. During the writing phase, negative or positive pulse is applied for a specific period of time, depending on the error (y) received from the difference between the desired value and the

Table 1. Variables definitions used with the memristor.

Variable	Definition
e	Control signal
y	Error signal
s	Memristor state
α	Pulse width mismatch

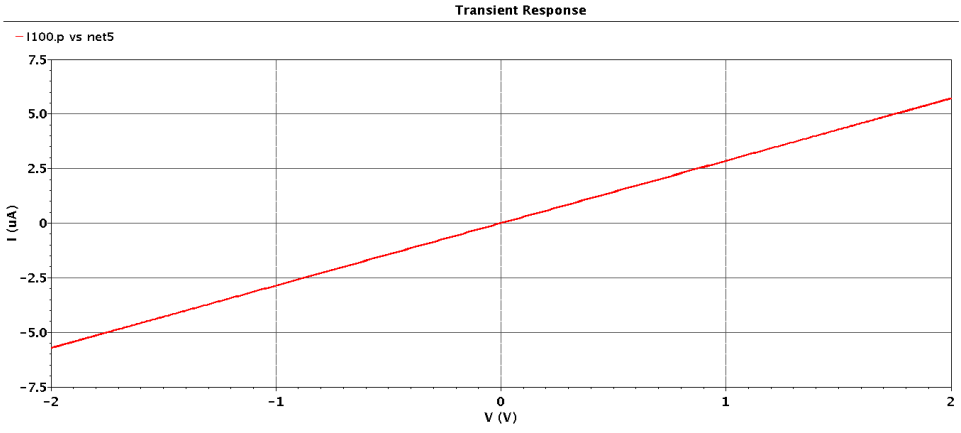

Fig. 2. I–V characteristics of the proposed synapse.

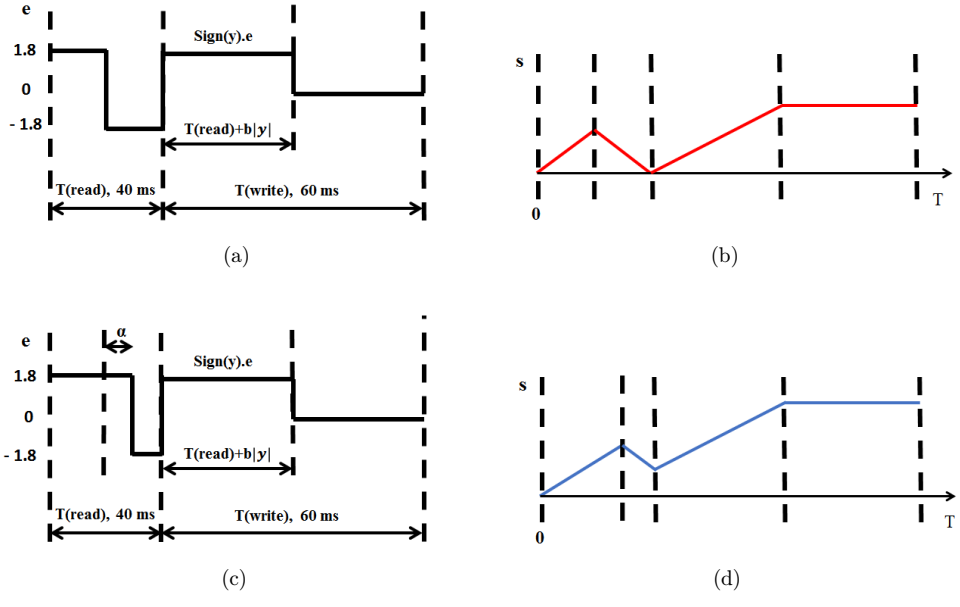

Fig. 3. A full period (100 ms) of the control signal e for both read and write of synapse weight. (a) The original signal, (b) the state signal corresponding to the original synapse, (c) signal with pulse width mismatch α where y, the error signal, is the difference between the desired output and the calculated output and (d) the change in state reading weight due to the mismatch.[17]

output value from the last reading following the weight update backpropagation algorithm. Correspondingly, the weight is modified based on the applied signal and the time duration.[10] Figure 3(b) displays the synapse weight value, represented by the memristor state during the reading phase with a symmetric square wave

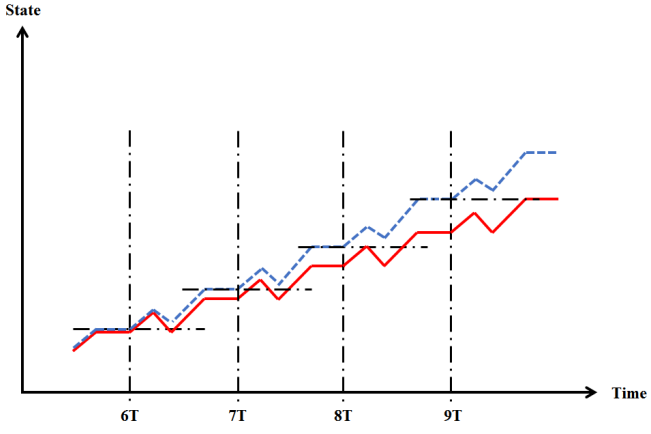

Fig. 4. Failure happened in writing weight after nine periods of operation due to reading signal mismatch. Solid line for original signal without mismatch and dashed line for signal with positive mismatch α.

(i.e., the memristor state change is maintained) and the writing phase with a specific pulse to update the weight

value (i.e., memristor state).

The adopted symmetric square wave in Ref. 10 is very efficient in simulations; however, it is not practical when applied in experimental measurements. This is because practically there is a mismatch between the positive and negative pulse duration.[17,18]

This pulse width mismatch between the negative and positive signals, as shown in Fig. 3(c), gives incorrect stored weight after several periods of operation.[18] Figure 3(d) shows the weight destructive reading operation in a single period. The failure of writing occurs when the accumulated weight error due to this mismatch is greater than or equal to the weight that should be written in the following phase without mismatch,[18] as displayed in Fig. 4. To compensate this error and get the correct weight value, a refreshment circuit is proposed in Sec. 3. In the previous work,[18] less destructive reading signals have been introduced, and, in this paper, a novel refreshment circuit is proposed to fulfill the circuit implementation of the idea proposed in Ref. 18.

3. Refreshment Circuit

The refreshment circuit shown in Fig. 5 fixes the failure that happens in the practical reading signal of synapse. The connections between this circuit and the synapse that suffer from mismatched signals are shown in Fig. 6.

The synapse without reading signal mismatch has been firstly implemented, and the state signal has been captured in a lookup table. This lookup table has been used to compare the captured signal with the state signal from the synapse suffering from

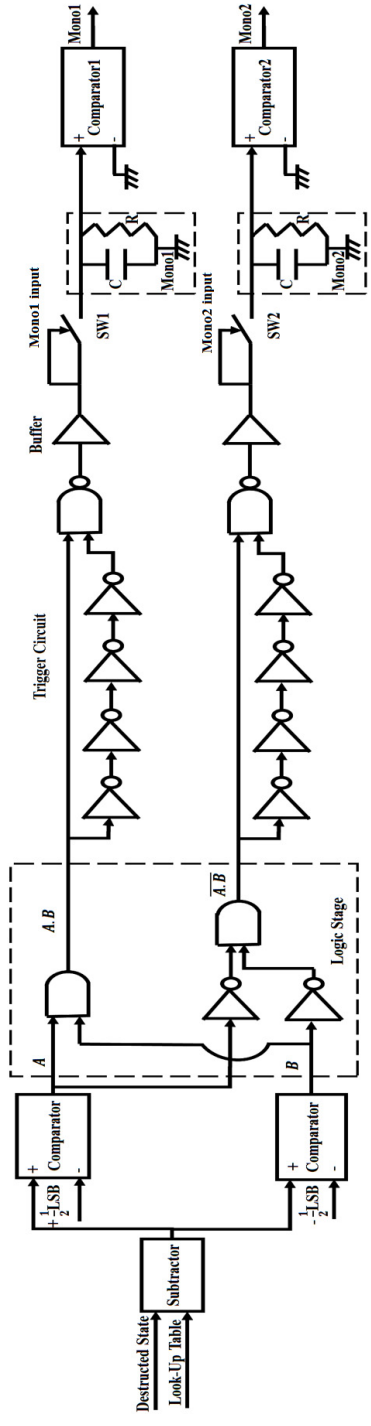

Fig. 5. The refreshment circuit block diagram.

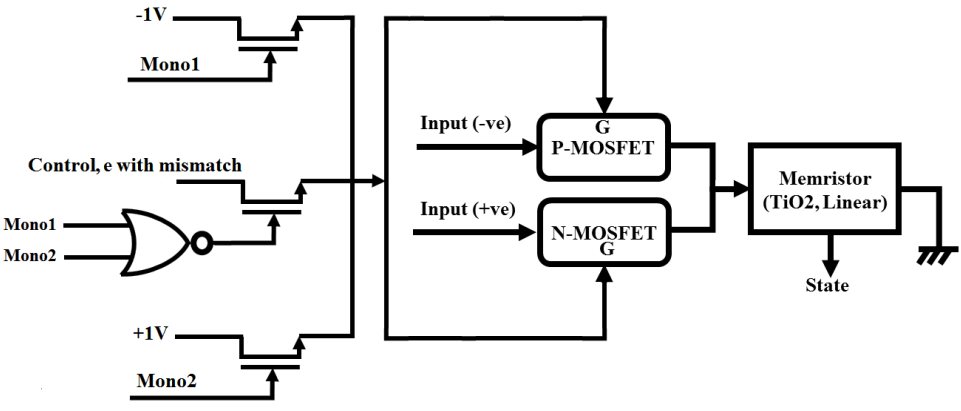

Fig. 6. New memristor-based synapse control connections due to mismatch in control e.

reading signal mismatch during operation. The comparison has been performed by the subtractor, as shown in the first stage of Fig. 5.

The subtractor output is the difference between the destructed state suffering from reading signal mismatch and the original state with ideal reading signal. This difference might have a positive or negative value depending on where the mismatch takes place in the reading signal (in positive half-cycle or negative one).

For example, if the mismatched signal has more width in the positive half-cycle than the negative one, as in Fig. 3(d), the subtractor output is positive. This means that the state suffering from mismatch will deviate in higher values than the original state signal, see Fig. 4.

The state with mismatch can have an increased or decreased value with respect to the original one and results in two cases of failure, positive and negative failures. Two comparators after the first subtractor are used to predict the error due to the mismatch before the failure. The failure here is represented as the least significant bit (LSB), which is the difference between the destructed state and the original one, when the destructed state value of signal (having positive mismatch α) in the 9th cycle reaches a higher value than it should be, which equals the same value of the original state (without mismatch) in the 10th cycle, see Fig. 4.

The comparison is made with half of the LSB ($\pm 150\,\mu$V), that can be changed according to the accuracy and speed required for the NN. One comparator is used to predict the positive error and the other one to predict the negative error, depending on the mismatch type mentioned earlier.

A logic stage is designed to monitor comparator outputs, A and B as shown in Fig. 5, and allow either Mono1 or Mono2 path to control synapse, as in Fig. 6, according to positive or negative error cases. The comparators have four output levels but only two cases are needed in our circuit, i.e., positive error (greater than $+1/2$ LSB) and negative error (less than $-1/2$ LSB). The output $A \cdot B$ represents the

error when positive mismatched signal is applied to the synapse, and $A^{-} \cdot B$ represents the error when negative mismatch is applied.

To refresh the value of the synapse weight caused by error, an interrupt is forced on the control signal of the synapse, as in Fig. 6. This is done by the monostable circuit.[20,21]

An RC monostable circuit is used here with SW1 and SW2 as triggers to activate Mono1 and Mono2 circuits, respectively. The resistance and capacitance of the monostable circuit take values of 80 KΩ and 5 nF, respectively.

The trigger signal is a 1 V pulse with 2 ns duration. To take this trigger signal from the steady-state logics $A \cdot B$ and $A^{-} \cdot B$, a delay circuit is used between the monostable circuit and the logic stage. This delay circuit is a simple logic delay with inverters that is compared with the original signal using a NAND gate. This gives the small pulse duration needed to trigger the monostable circuits. Two comparators, Comparator1 and Comparator2, are used to reduce the loading effect.

Figure 6 represents the synapse suffering from mismatch in the control signal e, with the new control connections from the refreshment circuit outputs, Mono1 and Mono2 of Fig. 5. If a positive error is detected by the proposed refreshment circuit, Mono1 is activated and applied to write a negative voltage as a correction to the destructed value. All the other control connections are deactivated, as Mono2 is zero. On the other hand, if a negative error is detected, a positive voltage is applied by Mono2 control. All the other control connections are deactivated, as Mono1 is zero. If no error has been detected (the difference between the state signals is less than $+150\,\mu$V and greater than $-150\,\mu$V), both Mono1 and Mono2 are deactivated, and the typical control signal e is applied to the synapse.

4. Simulation Verifications

Simulation of the refreshment control circuit, as shown in Figs. 5 and 6, has been performed by using Cadence Virtuoso with hardware-calibrated CMOS TSMC 130 nm technology for all transistors. The memristor Verilog-A linear ion drift model in Ref. 19 is used. All components and signal parameters of the original synapse and synapse with a reading pulse width mismatch are stated in Table 2.[18] Transistors stated in Table 2 are for the synapse configuration in Fig. 1. The cycle time T used for read and write operations equals 100 ms [1].[1] All switches are designed by practical transistors with the parameters shown in Table 3 to ensure a practical simulation. SW1 and SW2 are N-type metal-oxide-semiconductor (NMOS) transistors.

To make sure that the refreshment circuit has no impact on the synapse outputs calculated in Ref. 18, different mismatch percentages are tested. Pulse width mismatch α equals $200\,\mu$s (1%) gives a failure after 21 cycles.[18] α which equals $400\,\mu$s (2%) gives a failure after 19 cycles, as in Ref. 18. It is obvious that increasing the mismatch α results in early failures. Table 4 states the number of cycles to fail at different mismatches up to 5%.

Table 2. Circuit parameters.

	Parameter	Description	Value	Units
Memristor	R_{off}	Higher resistance	350	KΩ
	R_{on}	Lowest resistance	100	Ω
	D	Length	10	Nm
	μ_v	Linear ion mobility	10^{-14}	$m^2/\text{s}\cdot\text{V}$
Scaling	a	Input scaling	1	mV
	b	Error timing conversion	0.6T	S
Timing	T	Period	0.1	S
	T_{read}	Reading period	40	Ms
	T_{write}	Writing period	60	Ms
Input	X	Data input	10	—
Control signal	e	Control signal amplitude	±1	V
Transistors of the synapse	L	Length	130	nm
	W	Width	NMOS: 2 PMOS: 2	μm

Table 3. Parameters of all NMOS switches.

	Parameter	Description	Value	Units
SW1 and SW2 before the monostable stage	L	Length	130	nm
	W	Width	800	μm
NMOS Switches controlling refresh of synapse with mismatch	L	Length	130	nm
	W	Width	16 30 46	μm

Table 4. Number of cycles till failure happens.

α (%)	No. of cycles to fail (T)
1	21
2	18
3	16
4	14
5	13

Verification of the refreshment circuit function is shown in Figs. 7–11. Figure 7 shows the consequence of the outputs at different stages in Fig. 5 in the case of a positive failure (difference between the two state signals $\geq +150\,\mu$V). The output from the comparator stages is illustrated in first two plots of Fig. 7 ($A = 1$). The next step is the logic state giving $A \cdot B = 1$, indicating that a Mono1 pulse is generated to refresh the circuit. Finally, the output from the buffer stage is the trigger to enable Mono1.

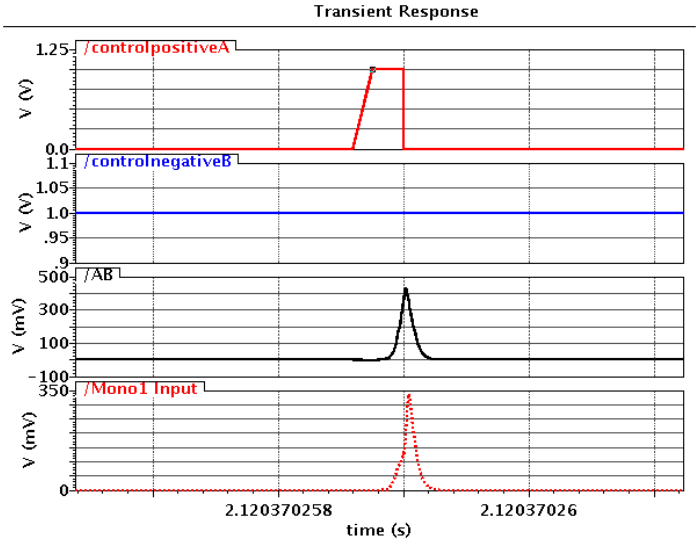

Fig. 7. The output $\boldsymbol{A} \cdot \boldsymbol{B}$ signals from the comparator stage; then $\boldsymbol{A} \cdot \boldsymbol{B}$ signals from the logic stage; then the indicator (the trigger pulse) gives a Mono1 pulse to refresh the weight stored in the synapse.

Figure 8 illustrates the weight being restored to its original value when the failure takes place, based on mismatch $\alpha = 1\%$. Mono1 pulse is generated to refresh the circuit, and during this duration, the error difference between the destructed state value and the reference value at the lookup table goes to zero.

Figure 9 illustrates the weight being restored to its original value when the failure takes place, based on mismatch $\alpha = -1\%$. Mono2 pulse is generated to refresh the circuit, and during this duration, the error difference between the destructed state value and the reference value at the lookup table goes to zero.

Another verification is shown in Figs. 10 and 11, where positive and negative mismatches $\alpha = \pm 5\%$ are applied, respectively. As shown, the error difference between the destructed state value and reference value at the lookup table goes to zero during Mono1 and Mono2 durations, respectively.

Figures 8–11 indicate that the proposed circuit achieves the task of refreshment for both positive and negative mismatches with different values.

Figure 12 is another verification of the performance for the proposed circuit, where three consecutive mismatch values are applied in the control signal e. The proposed circuit succeeded in refreshment task corresponding to type of failure (positive or negative).

5. The PVT Performance

The proposed design performance of different corners is still the same or even enhanced. Table 5 shows the circuit performance for process variation (TT, (typical

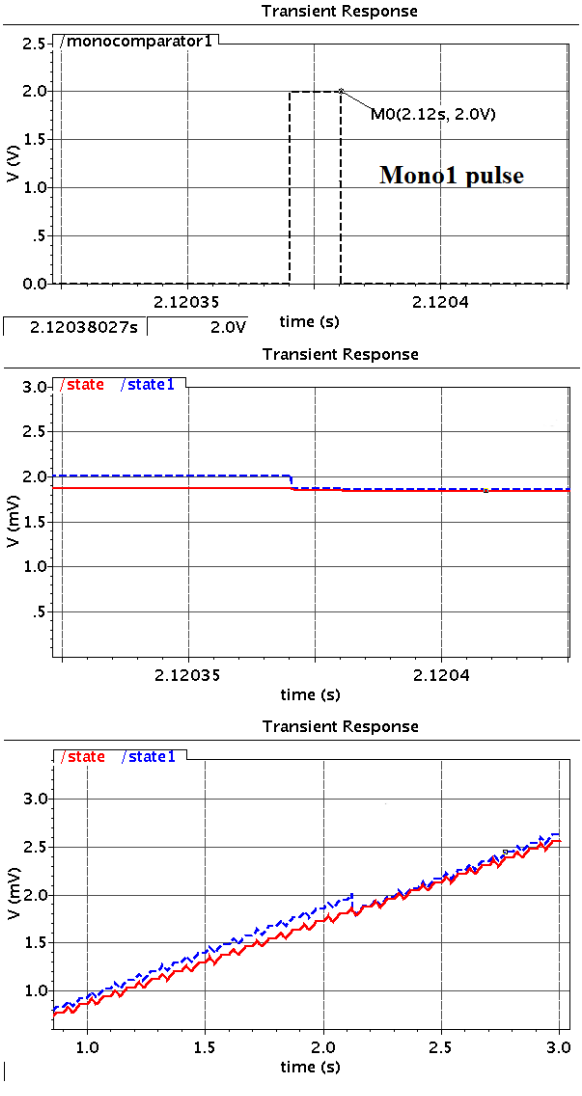

Fig. 8. The Mono1 pulse to refresh the circuit the top plot, the two state signals (zoomed in) in the middle plot and (zoomed out) in the bottom plot. Solid line represents original signal without mismatch and dashed line represents signal with positive mismatch $\boldsymbol{\alpha} = \mathbf{1\%}$.

typical), SS (slow slow), SF (slow fast), FS (fast slow) and FF (fast fast) corners) and for voltage variation (0.9, 1 and 1.1 V) and finally for temperature variation (-40, 27 and 85°C).

6. A Simple 2×2 Synaptic Grid Circuit

The proposed synapse array was tested using CMOS TSMC 130 nm and linear ion drift memristor model.[9,10] A small 2×2 synaptic grid circuit,[10] as shown in Fig. 13, is

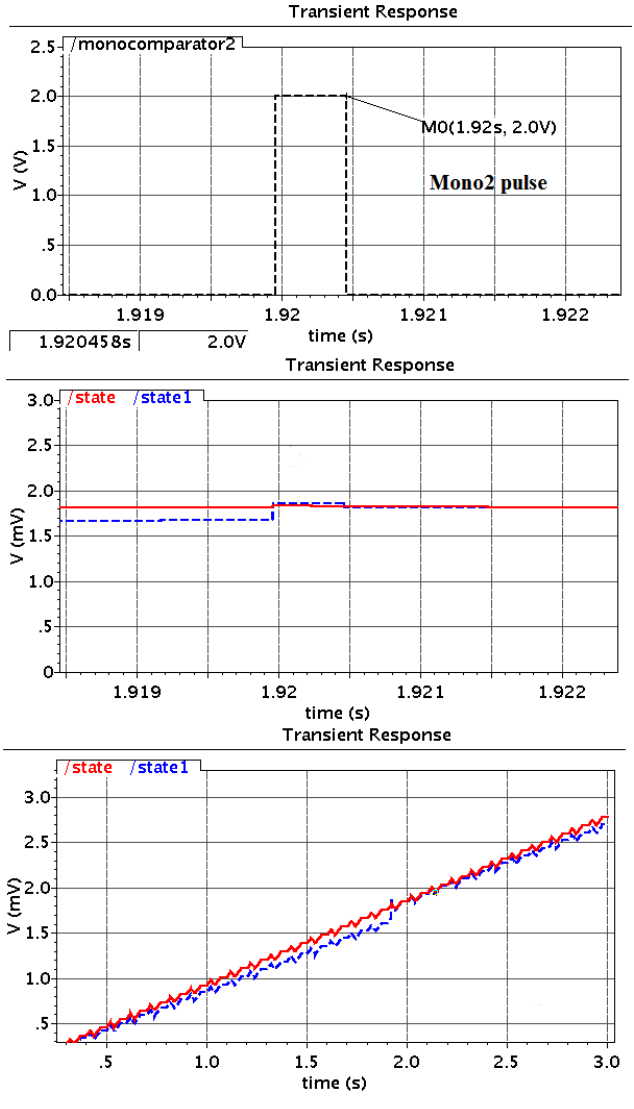

Fig. 9. The Mono2 pulse to refresh the circuit in the top plot, the two state signals (zoomed in) in the middle plot and (zoomed out) in the bottom plot. Solid line represents original signal without mismatch and dashed line represents signal with negative mismatch $\boldsymbol{\alpha} = -\mathbf{1\%}$.

simulated for time 10 T (10 read–write cycles) with a simple input:

$$(x_1, x_2) = (1, -2) * 10\,\mathrm{sign}(t - 5\mathrm{T}),$$
$$(y_1, y_2) = (0.5, -0.25).$$

Those inputs are shown in Figs. 14 and 15; the memristor voltages and the states are also shown. The basic operation of the circuit is verified.

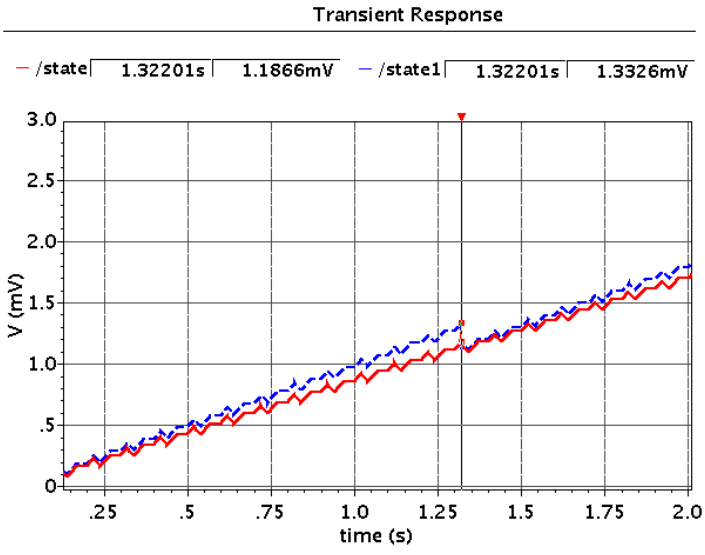

Fig. 10. Difference increased between the original state signal (solid line) and the state signal that suffers from positive mismatch $\boldsymbol{\alpha} = \mathbf{5\%}$. (dashed line). Then, the difference goes to zero after refreshment.

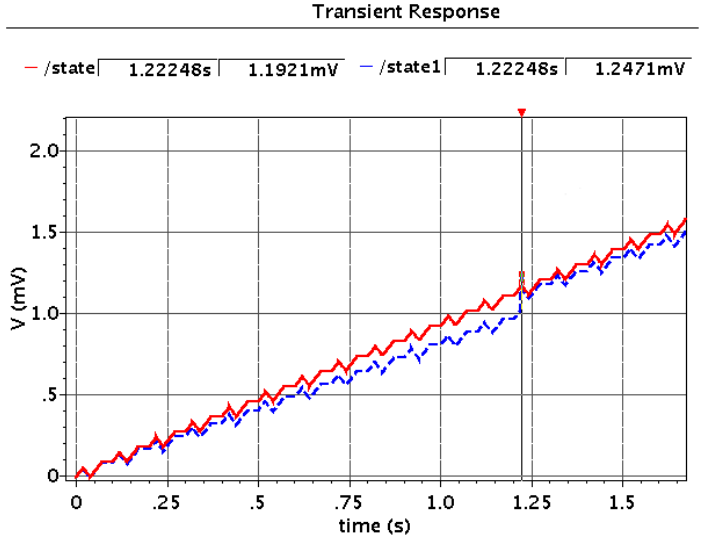

Fig. 11. Difference decreased between the original state signal (solid line) and the state signal that suffers from negative mismatch $\boldsymbol{\alpha} = \mathbf{-5\%}$. (dashed line). Then, the difference goes to zero after refreshment.

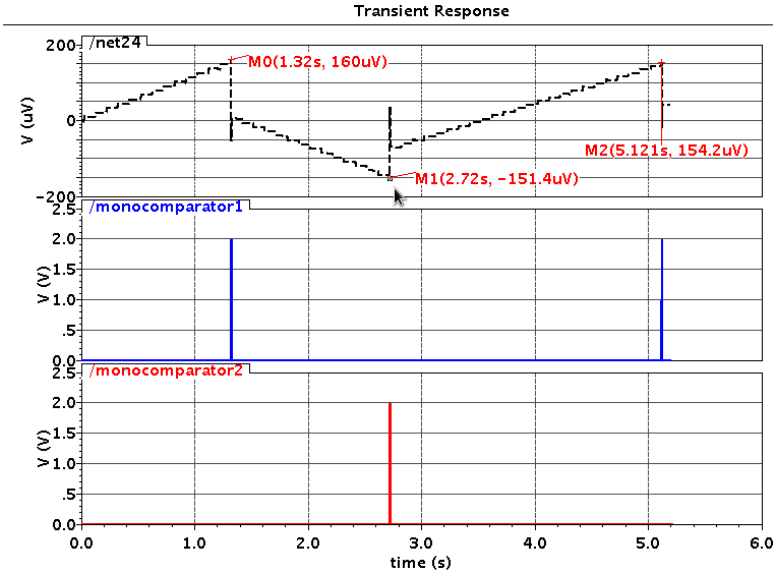

Fig. 12. Different consecutive mismatch values start with positive mismatch ($\boldsymbol{\alpha = 5\%}$), followed by negative mismatch ($\boldsymbol{\alpha = -4\%}$), and end with a different positive mismatch ($\boldsymbol{\alpha = 3\%}$).

Table 5. The PVT performance.

	Corner	No. of cycles before refreshment is needed (T)
Process	TT state	21
	SS corner	22
	SF corner	23
	FS corner	21
	FF corner	21
Voltage	0.9 V	22
	1 V	21
	1.1 V	21
Temperature	−40°C	22
	27°C	21
	85°C	22

7. Comparison to Other Works

Investigating the read and write process, the read operation was not found as simple as the write operation. If the read mechanism is not well designed, the memristor state may be distracted and error can occur after multiple read cycles.

So, many researchers studied this case for different memristors and determined how many cycles are needed before failure would happen and if there is a possible solution to compensate this problem.

Table 6 concludes a comparison between this research and other ones.

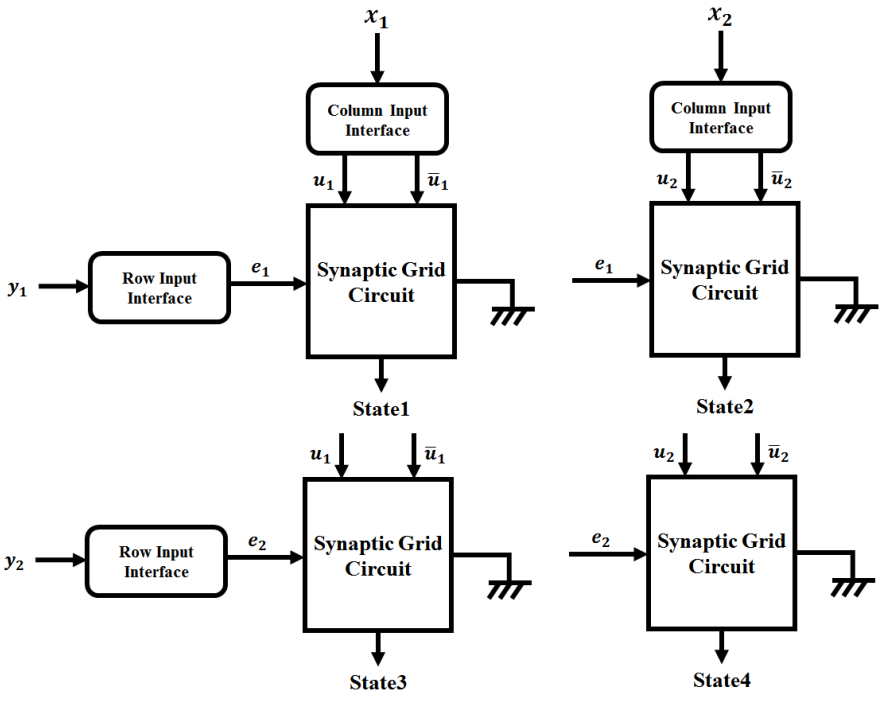

Fig. 13. A simple 2×2 synaptic grid circuit.

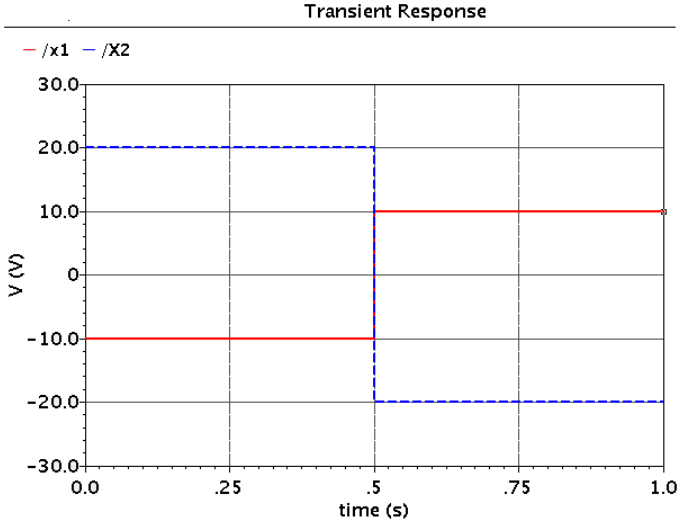

Fig. 14. Inputs $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ of the synaptic 2×2 grid circit.

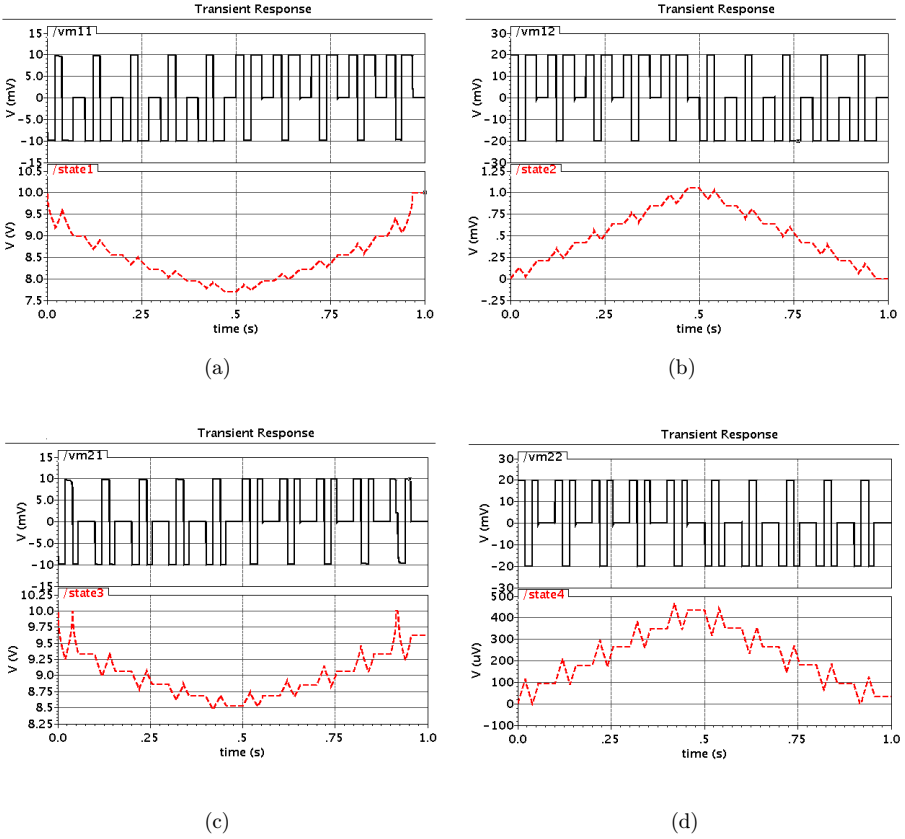

Fig. 15. The voltage V upon the memristor and the conductance (the state) of the memristor are shown, respectively, by the black solid and red dashed lines: (a) $\boldsymbol{V}_{11}$ and state 1, (b) $\boldsymbol{V}_{12}$ and state 2, (c) $\boldsymbol{V}_{21}$ and state 3 and (d) $\boldsymbol{V}_{22}$ and state 4 (color online).

Table 6. Comparison.

	This work	Reference 17	Reference 22	Reference 23	Reference 24	Reference 25
Memristor model	Linear iondrift	Memristor model in Ref. 9	The spintronic memristor	Linear iondrift	Memristor model in Refs. 26 and 27	VTEAM
Transistor technology	TSMC 130 nm	No transistors	No transistors	Gpdk180 library for opamp	TSMC 130 nm	No transistors
No. of cycles before refresh is needed	21 cycles	19 cycles	60 cycles	—	—	—
Refreshment circuit exist or not	Proposed refreshment circuit	—	—	—	—	Refreshment after logic 1 only
Power dissipation	Average power 14.42 mW	—	Average write power 280 μW Average read power 71 μW	—	Average write power 1.88 mW Average read power 0.916 μW	—

8. Conclusion

In this paper, a novel refreshment circuit for 2T1M neuromorphic synapse has been illustrated. The circuit gives a novel method to compensate the natural mismatch occurring in the generated square pulses that could destruct the stored weight after many cycles of operation. The circuit achieves the task with very small interrupt time. The interrupt is locally placed, which means no need to disconnect the synapse from the NN to be refreshed. And the basic operation of the memristor is verified by building a simple 2×2 grid circuit.

Future work will apply this circuit with different memristor material types that are used in 2T1M synapse. Also, collaboration with other transistor–memristor synapses will be investigated.

9. Acknowledgment

This work was partially funded by ONE Lab at Zewail City of Science and Technology and Cairo University, NTRA, ITIDA, ASRT, Mentor Graphics, NSERC.

References

1. O. Krestinskaya, A. P. James and L. O. Chua, Neuromemristive circuits for edge computing: A review, *IEEE Trans. Neural Netw. Learn. Syst.* **2** (2019) 4–23.
2. Y. Jiang, C. Yang, J. Na, G. Li, Y. Li and J. Zhong, A brief review of neural networks based learning and control and their applications for robots, *Complexity* **2017** (2017) 1–14.
3. I. Lee and K. Lee, The internet of things (IoT): Applications, investments, and challenges for enterprises, *Bus. Horiz.* **58** (2015) 431–440.
4. S. Madakam, R. Ramaswamy and S. Tripathi, Internet of things (IoT): A literature review, *J. Comput. Commun.* **03** (2015) 164–173.
5. C. Frenkel, M. Lefebvre, J.-D. Legat and D. Bol, A 0.086-mm^{2}12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28 nm CMOS, *IEEE Trans. Biomed. Circuits Syst.* **13** (2019) 145–158.
6. N. Aloysius and M. Geetha, A review on deep convolutional neural networks, *Int. Conf. Communication and Signal Processing (ICCSP)*, Chennai, India, April 2017, pp. 588–592.
7. J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai and T. Chen, Recent advances in convolutional neural networks, *Pattern Recognit.* **77** (2018) 354–377.
8. M. Pfeiffer and T. Pfeil, Deep learning with spiking neurons: Opportunities and challenges, *Front. Neurosci.* **12** (2018) 774–791.
9. D. B. Strukov, G. S. Snider, D. R. Stewart and R. S. Williams, The missing memristor found, *Nature* **453** (2008) 80–83.
10. D. Soudry, D. Di Castro, A. Gal, A. Kolodny and S. Kvatinsky, Memristor-based multilayer neural networks with online gradient descent training, *IEEE Trans. Neural Netw. Learn. Syst.* **26** (2015) 2408–2421.
11. W.-H. Chen, W.-S. Khwa, J.-Y. Li, W.-Y. Lin, H.-T. Lin, Y. Liu, Y. Wang, H. Wu, H. Yang and M.-F. Chang, Circuit design for beyond von Neumann applications using emerging memory: From nonvolatile logics to neuromorphic computing, *IEEE 18th Int. Symp. Quality Electronic Design (ISQED)*, Pisctataway, NJ, 2017, pp. 23–28.

12. H. A. Du Nguyen, J. Yu, L. Xie, M. Taouil, S. Hamdioui and D. Fey, Memristive devices for computing: Beyond CMOS and beyond von Neumann, *IEEE Int. Conf. Very Large Scale Integration (VLSI-SoC)*, Abu Dhabi, United Arab Emirates, October 2017, pp. 1–10.
13. M. R. Azghadi, B. Linares-Barranco, D. Abbott and P. H. W. Leong, A hybrid CMOS-memristor neuromorphic synapse, *IEEE Trans. Biomed. Circuits Syst.* **11** (2016) 434–445.
14. Y. Zhang, X. Wang and E. G. Friedman, Memristor-based circuit design for multilayer neural networks, *IEEE Trans. Circuits and Syst. I: Regul. Pap.* **65** (2018) 677–686.
15. M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev and D. B. Strukov, Training and operation of an integrated neuromorphic network based on metal-oxide memristors, *Nature* **521** (2015) 61–64.
16. M. Hu, Y. Chen, J. J. Yang, Y. Wang and H. H. Li, A compact memristor-based dynamic synapse for spiking neural networks, *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **36** (2017) 1353–1366.
17. Y. Ho, G. M. Huang and P. Li, Dynamical properties and design analysis for nonvolatile memristor memories, *IEEE Trans. Circuits Syst. I: Regul. Pap.* **58** (2011) 724–736.
18. A. H. Hassan and H. Mostafa, Nondestructive reading and refreshment circuit for memristor-based neuromorphic synapse, *IEEE 62nd Int. Midwest Symp. Circuits and Systems (MWSCAS)*, Dallas, TX, USA, August 2019, pp. 642–645.
19. S. Kvatinsky, K. Talisveyberg, D. Fliter, A. Kolodny, U. C. Weiser and E. G. Friedman, Models of memristors for SPICE simulations, *IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, Eilat, November 2012, pp. 1–5.
20. C. F. Ho, Zero Quiescent Current Monostable Multivibrator, *Radio Electron. Eng.* **37** (1969) 22–24.
21. R. C. Bowes, A new linear delay circuit based on an emitter-coupled multivibrator, *Proc. IEEE B: Electron. Commun. Eng.* **106** (1959) 793–800.
22. S. F. Nafea, A. A. S. Dessouki, S. El-Rabaie, B. E. Elnaghi, Y. Ismail and H. Mostafa, Area-efficient read/write circuit for spintronic memristor based memories, *IEEE 60th Int. Midwest Symp. Circuits and Systems (MWSCAS)*, Boston, MA, USA, August 2017, pp. 1544–1547.
23. M. G. S. P. Reddy and V. Ravi, Nondestructive read circuit for memristor-based memories, *Nanoelectronic Materials and Devices*, Lecture Notes in Electrical Engineering (2018), pp. 123–131.
24. A. A. Daoud, A. S. Dessouki, H. Mostafa, R. M. Abdallah and E.-S. M. El-Rabaie, A high precision write/read circuits for memristors using digital input/output interfaces, *Microelectron. J.* (2020).
25. V. Ravi, K. Chitra and S. R. S. Prabaharan, Variation-tolerant, low-power, and high endurance read scheme for memristor memories, *Analog Integrated Circuits and Signal Processing* (Springer, 2020), pp. 83–98.
26. A. A. Daoud, A. A. S. Dessouki and S. M. Abuelenin, Quantitative analysis of memristance defined exponential model for multi-bits titanium dioxide memristor memory cell, *Adv. Electr. Comput. Eng.* **16** (2016) 75–84.
27. A. A. Daoud, A. A. S. Dessouki and S. M. Abuelenin, An accurate memristor model based on a novel definition of memristance, *IEEE 28th Int. Conf. Microelectronics (ICM)*, December 2016, pp. 61–64.