Article

Not peer-reviewed version

A Novel Programming Circuit for Memristors

Shengtao Tu , Jinyu Li , Yanyun Ren , Qin Jiang , Shisheng Xiong [*]

Posted Date: 31 May 2023

doi: 10.20944/preprints202305.2195.v1

Keywords: memristor; memristor programmer circuit; multi-level memristance; memristor crossbar

Article

A Novel Programming Circuit for Memristors

Shengtao Tu [1], Jinyu Li [1], Yanyun Ren [2], Qin Jiang [1] and Shisheng Xiong [1,*]

1. State Key Laboratory of ASIC and System, Micro Nano System Center, School of Information Science and Technology, Fudan University, Shanghai 200433, P. R. China China
2. 2020 X-Lab, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Science, Shanghai 200050, China

* Correspondence: sxiong@fudan.edu.cn (S.S.X)

Abstract: Memristor has attracted a lot of interest due to its high processing speed, low power consumption and high integration ability, which is critical for electronic systems and memory-centric computing. However, the memristor programming circuit and strategy are still inflexible and complex, since the signal generator/collector and stimulate pulse must be carefully matched and designed based on memristor intrinsic characteristics without reconfigurable. Here, a simple and effective circuit only consists a parallel reference-resistor-and-NMOS is designed to program memristor with a more than 99% memristance precision. And the amplitude and width of stimulate pulse are fixed to ±4V and 5ms, respectively. In order to cope with the device variation, such as ±10% tolerance of transition voltage, an optimized programming strategy was proposed and demonstrated great robustness. Additionally, a set of reference resistors and NMOSs have been added to facilitate multi-level memristance operation without requiring any changes to the circuit structure. This program circuit was also employed to program memristor crossbar remains 99% precision. In the end, a memristor-based convolutional neural network which controlled by our optimized programming circuit was used for image recognition, and 89.36% accuracy can be achieved even under 15.8% memristance tolerance. This novel circuit demonstrates a simple and flexible strategy in memristor programming, providing a new way to control memristor crossbar for practical application.

Keywords: memristor; memristor programmer circuit; multi-level memristance; memristor crossbar

1. Introduction

In 1971, the memristor is the fourth basic circuit element that is proposed by L. O. Chua in theory 1. . As the fourth basic circuit element besides resistor, capacitor, and inductor, memristor has been considered as one of the next generations of non-volatile memory technology. The memristor is a nonvolatile device and has variable memristance. When the current flows through the memristor from different directions, the memristance will change accordingly. When the current cross the memristor was removed, the memristance will remained. In 2008, the first physical memristor was reported by HP labs 2. .The phenomenon of resistive switching in electronic devices has attracted wide interest from academia and industry, due to its application to store information in a memristor, enabling the development of neuromorphic and memory-centric computing systems 3. .

In recent years, programmable and reconfigurable analog elements have attracted considerable attention. In the literature, there have been various studies on the programmable circuit. In 2010, Pershin et al. developed a programmable memristor circuit that includes a microprocessor, analog-to-digital converter (ADC), and digital potentiometer 10. . In 2012, Berdan et al. presented a circuit that exploited the dynamic modulation of resistance under a constant DC bias on the operation of an analog programming circuit for accurately setting the state of a memristor 11. .In 2016, Merced et al. investigated memristor 1T1R arrays rapidly reaching arbitrary conductance states and programming was performed by applying an adaptive pulsed algorithm that utilized the transistor gate voltage to control the SET switching operation12. . Kim et al. suggested an interesting tuning approach that is to exploit the voltage divider (VD) effect and its HW simplicity via a resistor in series with the memristor 13. . In 2017, Olumodeji et al. presented a circuit for accurately programming the memristor in both an incremental and a decremental fashion using auto-tuning operational amplifier's gain, which exploited the characteristics of the memristor, and this circuit took advantage of the memristor's pulse-based programmability 16. . Mokhtar et al. presented a pulse-coded memristor programming method adopted in RWC (resistance writing circuit) design. It used 2 sets of switches to decrease or increase memristance by supplying a positive or negative pulse 17. . In 2020, Tarkhan et al. presented a novel CMOS circuit for programming memristors which used a Wheatstone bridge circuit to measure the current memristance while the programming current was flowing through the device 18. . In 2022, Knowm Inc. explored different circuit topologies and approaches to perform the forming of RRAM and a target-resistance was pursued through pulsed voltage stress, followed by cycle-to-cycle stabilization using a custom trans-impedance amplifier circuit 19. . In 2022, Lu et al. analyzed the reason of the resistance state deviation of the memristor by studing predecessors on read and write circuit, and proposed a novel structure of memristor based on opposite polarity 20. . In 2023, Randrianantenaina et al. reported programmable and reconfigurable analog/digital platforms as an alternative perspective in order to promote analog and digital applications based on memristors 21. .

In this paper, we suggest an approach to program memristors in analog circuits based on the threshold-type behavior of the memristor. Our main idea is to use a negative source to programs the memristor to high resistance states (HRS). The positive source then programs the memristor to aim memristance. In addition, we have added a reference resistor to program the memristor to multi-level memristance according to digital input signals. Finally, a convolutional neural network (CNN) has been trained by our circuit and 77.64% classification precision can be achieved even under 22.2% memristance tolerance. In this way, we obtain a programmable circuit that can obtain accurate memristance and has a simple structure.

2. The memristor Programmable Circuit

2.1. Memristor Model

The adopted memristor model has been proposed by Pershin Y. V. 22. . The memristor has a threshold characteristic. In other words, when the voltage applied to the memristor is greater than the threshold value, the memristance (memristor resistance) will change. Otherwise, the memristance will remains unchanged. The mathematical expressions of this model are as follows:

$$I(t) = R_M^{-1}(X, V_M, t) * V_M(t) \tag{1}$$

$$\dot{X} = f(X, V_M, t) \tag{2}$$

$$I = X^{-1} * V_M \tag{3}$$

$$\frac{dX}{dt} = f(V_M)[\theta(V_M)\theta(R_{off} - X) + \theta(-V_M)\theta(X - R_{on})] \tag{4}$$

$$f(V_M) = \beta V_M + 0.5(\alpha - \beta)[|V_M + V_t| - |V_M - V_t|] \tag{5}$$

where X represents n-th internal state variables, and X is a vector. $V_M(t)$ is the voltage applied to the memristor, *I(t)* is the current that flows through the memristor. V_t is the memristor threshold voltage. R_M is the memristance which is a scalar. R_{on} is the LRS and R_{off} is the HRS. The $\theta(\bullet)$ are step functions that are used to limit the memristance to the region between R_{on} and R_{off}. The coefficients α and β define the slopes of the $f(V_M)$ curve below and above the threshold, which characterize the rate of memristance change at $|V_M| < V_t$ and $|V_M| > V_t$, respectively.

A positive or negative voltage applied to the memristor decreases or increases the memristance R_M. The device state changes only when $|V_M| > V_t$ (for set transition $V_t = V_{set}$, and for reset transition $V_t = V_{reset}$). As shown in Figure 1., when the voltage value applied to the memristor is greater than the positive threshold of the memristor ($V_M > V_{set}$), the memristance decreases sharply, memristor tends to change from HRS to LRS, and the process is SET transition, when the voltage value applied to the memristor is less than the negative threshold of the memristor ($V_M < V_{reset}$), the memristance increases sharply, memristor tends to change from LRS to HRS, and the process is RESET transition.

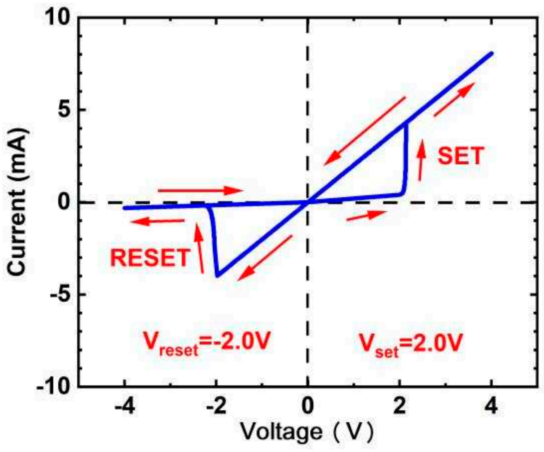

Figure 1. Simulation *I-V* curve of the memristor. V_{set} is the positive threshold with the value of 2.0V. When the voltage applied to the memristor is greater than 2.0V, the memristance decreases to R_{on} sharply. V_{reset} is the negative threshold with the value of -2.0V. When the voltage applied to the memristor is less than -2.0V, the memristance increases to R_{off} sharply.

2.2. The Memristor Programmable Circuit

In the programmable analog circuits of memristors, Pershin et al. developed a circuit that applied negative voltage to the memristors and then use positive voltage to program the memristor's states during their operation as analog circuit elements10. . In addition, the voltage divider effect is usually used in programmable circuit13. . This paper proposes a novel circuit by utilizing before design ideas and adding a reference resistor that is designed to be controllable. The circuit is simple and the memristance is configurable.

When programming the memristor, if the memristance is greater than the aim value, it is necessary to decrease the memristance through a SET process. If the memristance is less than the aim value, it is necessary to increase the memristance through a RESET process. Before programming the memristor, it is first to obtain the greatest memristance value R_{off} by RESET process. Then it programs the memristor to aim value. In addition, we also add a reference resistor which contributes to obtaining aim memristance, through this structure the memristance is configurable. The memristance is related to the reference resistor, and the reference resistor is controlled by digital input signals. The proposed memristor programmable circuit structure contains one memristor, one NMOS, and one reference resistor as shown in Figure 2. R_M is the memristor, R is the reference resistor, and M is the NMOS, respectively.

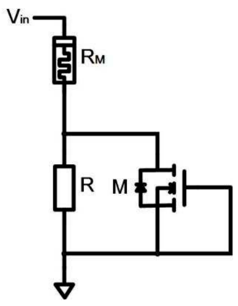

Figure 2. Memristor programming circuit. R_M is the memristor that to be programmed, R is the reference resistor and M is the NMOS.

The programming progress contains two steps:

Step 1: the input source $V_{in}(|V_{in}| > |V_{reset}|)$ is negative. the current flows through NMOS (NMOS can be regarded as a diode) and then through memristor R_M to reset the memristor, as shown in Figure 3. So, the memristance increases sharply to R_{off}.

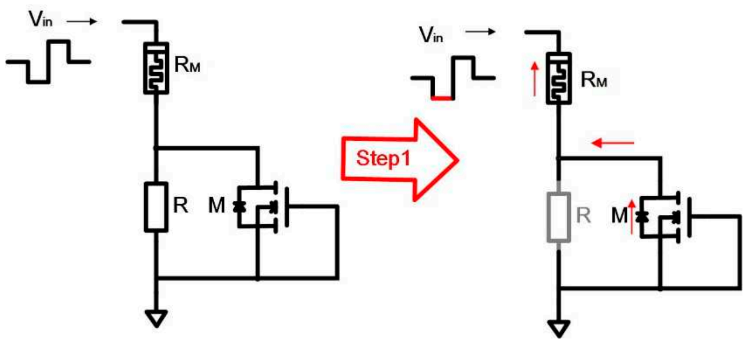

Figure 3. Step 1 of the memristor programming process. In this period, input source V_{in} is negative. The current flows through M and R_M, as the red arrows in the picture. And the voltage applied to the memristor is negative.

Step 2：the input source V_{in} is positive, and the current flows through memristor R_M and reference resistor R (NMOS can be regarded as open), as shown in Figure 4. The voltage applied to the memristor is positive (it must match $|V_{in}| > |V_{set}|$), and the voltage value is

$$V_M = \frac{R_M}{R_M + R} * V_{in} \tag{6}$$

when $V_M > V_{set}$, memristor SET, the memristance decreases gradually, and the memristor voltage value also decreases gradually, until $V_M = V_{set}$. When $V_M = V_{set}$, the memristor stops SET, so the memristance does not change and remains a certain value, the memristance value is

$$R_M = \frac{V_{set}}{V_{in} - V_{set}} * R \tag{7}$$

where V_{in} is the input source, V_{set} is the memristor SET transition voltage, R is the reference resistor.

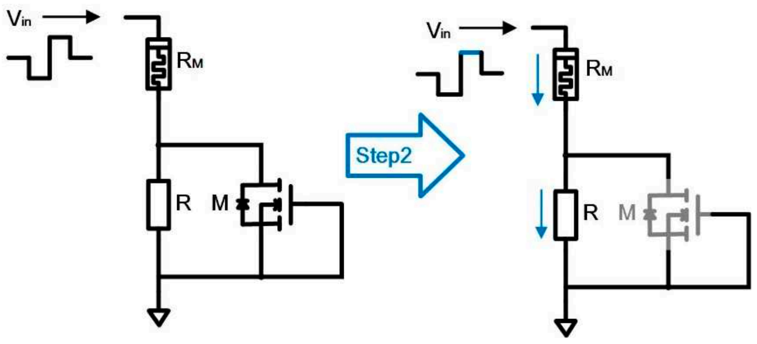

Figure 4. Step 2 of the memristor programming process. In this period, input source V_{in} is positive. The current flows through R_M and R, as the blue arrows in the picture. And the voltage applied to the memristor is positive.

When $V_{in} = 2V_{set0}, R = R_0, V_{set} = V_{set0}$, the memristance value can be set to R_0 after the programmable process. Hence, if we want to program the memristor to R_0, we only need to set the reference resistor $R = R_0$.

2.3. The Simulation of the Memristor Programmable Circuit

Figure 5. shows the simulation result of the memristor programmable circuit. The width of the programming process is 10ms. The first 0-5ms is the period of Step 1, during which the memristor RESET and the memristance increases to R_{off}. And the following 5-10ms is the period of Step 2, during which the memristor SET and the memristance decrease to the aim value. When the reference resistor is $R = 1\text{k}\Omega$, the memristance simulation result is 0.9999kΩ, compared with reference resistor R, the accuracy of the simulation result is more than 99%. When the reference resistor is R= 8kΩ, the memristance simulation result is 8.0097kΩ; corresponding to a more than 99% simulation accuracy. The simulation result shows that the programmable circuit function is correct, and the circuit has high precision.

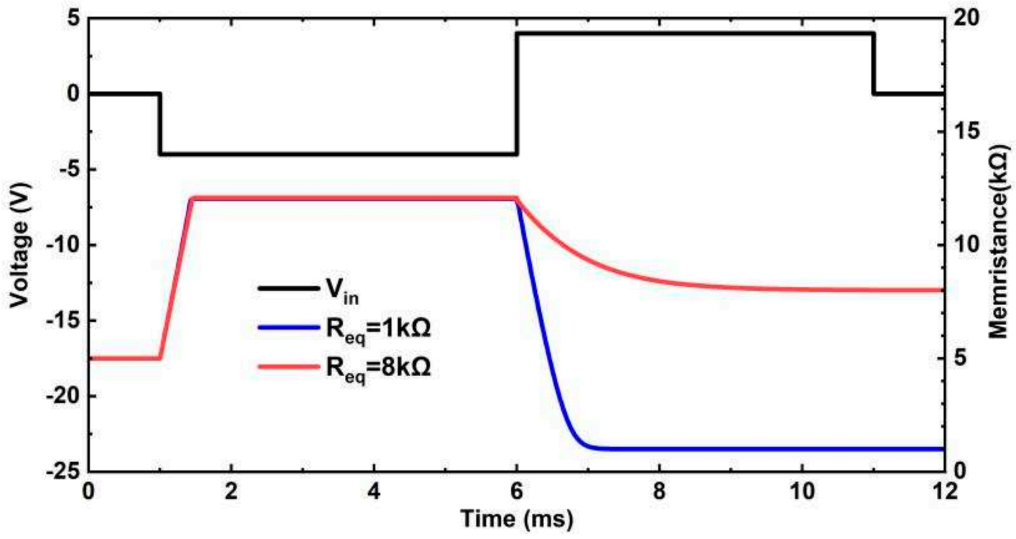

Figure 5. Simulation of the programmable circuit. The black line is the input source V_{in}, and the value is -4.0V during first 5ms and 4.0V during second 5ms of the pulse. The blue line represents the memristance when the reference resistor is R= 1kΩ. The red line represents the memristance when the reference resistor is R= 8kΩ.

3. Reference Resistor Control Circuit

3.1. The Reference Resistor Circuit

The reference resistor circuit is shown in Figure 6., which is composed of four NMOSs and four resistors. Different from the same type of NMOSs, the resistors are selected with 1kΩ、1kΩ、2kΩ、4kΩ. The four different resistors can achieve different R_{eq} values. The control signals are $\overline{b3}\,\overline{b2}\,\overline{b1}$, which are input digital signals. The reference resistor value is

$$R_{eq} = R_0 + R_1 * \overline{b1} + R_2 * \overline{b2} + R_3 * \overline{b3} \tag{8}$$

where R_0、R_1、R_2、R_3 are reference resistors; $\overline{b3}\,\overline{b2}\,\overline{b1}$ are input digital control signals.

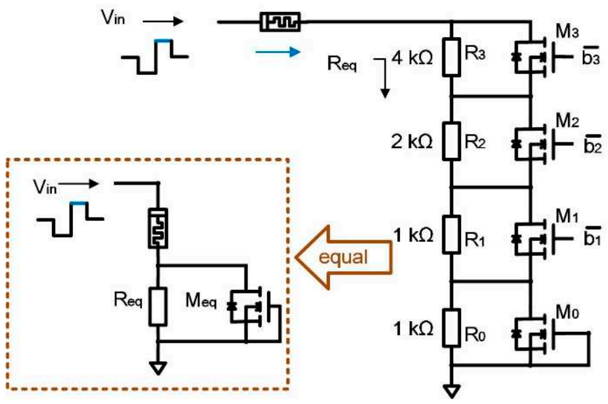

Figure 6. The reference resistor circuit with 3-input bits. R_0、R_1、R_2、R_3 are reference resistors; $\overline{b3}\,\overline{b2}\,\overline{b1}$ are digital control signals. The circuit is equivalent to the circuit diagram in the dashed box. R_{eq} is an equivalent resistance. M_{eq} is an equivalent NMOS.

The memristor programming circuit can be adjusted to be controlled by more digital input signals, thus the memristor could be programmed to multi-level states. As is shown in Figure 7. $\overline{bn}\ldots\overline{b1}$ are the control digital signals which can program memristor to multi-level memristance. The reference resistor value is

$$R_{eq} = R_0 + R_1 * \overline{b1} + \cdots + R_n * \overline{bn} \tag{9}$$

where R_0、R_1、… 、R_n are reference resistors; $\overline{b1}\ldots\overline{bn}$ are digital control signals;

doi:10.20944/preprints202305.2195.v1

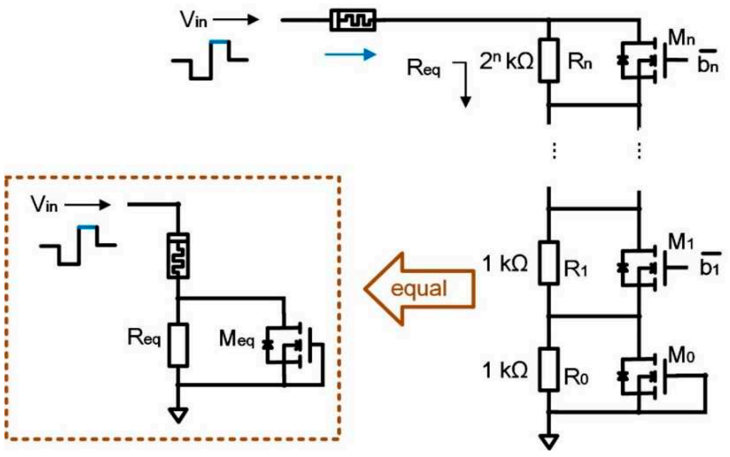

Figure 7. The reference resistor circuit with n-input bits. R_0、R_1、 … 、R_n are reference resistors; $\overline{bn}$ ... $\overline{b1}$ are digital control signals. The circuit is equivalent to a circuit diagram in the dashed box. R_{eq} is the equivalent resistance and M_{eq} is the equivalent NMOS.

3.2. The Simulation of the Reference Resistor Circuit

The simulation results are shown in Figure 8. The programming time cycle is 10ms. The first 5ms is the period of Step 1, and in this period the memristor can be reset to R_{off}. And the second 5ms is the period of Step 2, and in this period the memristor will be set to the aim value. When $\overline{b3}\,\overline{b2}\,\overline{b1} = 000\sim111$, the memristance simulation results are 1.0001kΩ、 2.0003kΩ、 3.0001kΩ、 4.0016kΩ、 5.0005kΩ、 6.0023kΩ、 7.0045kΩ、 8.0087kΩ, respectively. All of these simulation results keep 99% programming precision with R_{eq}.

Table 1. shows the relationship between $\overline{b3}\,\overline{b2}\,\overline{b1}$ and R_{eq}; The digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 000\sim111$, and the equivalent reference resistor is $R_{eq} = 1\text{k}\Omega\sim8\text{k}\Omega$, respectively. The memristance simulation results are also shown in Table 1.

Table 1. The simulation results based on different $\overline{b3}\,\overline{b2}\,\overline{b1}$ and R_{eq}.

$\overline{b3}\,\overline{b2}\,\overline{b1}$	R_{eq}	**Memristance**
000	1 kΩ	1.0001 kΩ
001	2 kΩ	2.0003 kΩ
010	3 kΩ	3.0001 kΩ
011	4 kΩ	4.0016 kΩ
100	5 kΩ	5.0005 kΩ
101	6 kΩ	6.0023 kΩ
110	7 kΩ	7.0045 kΩ
111	8 kΩ	8.0087 kΩ

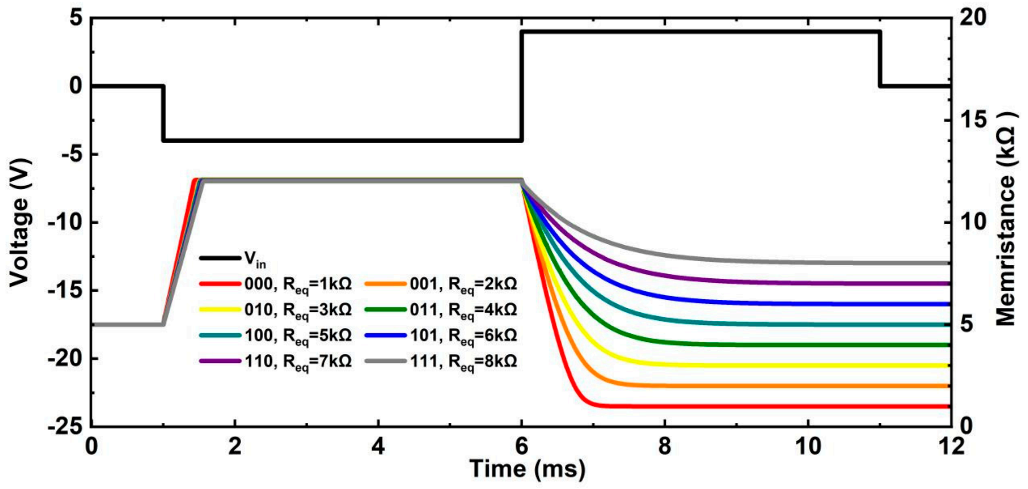

Figure 8. Simulation of the programmable circuit according to different input signals. The black line is the input source V_{in}. And the color lines are the memristances evolution as the function of different reference resistors.

4. Precision Analysis and simulation

From the formula *(7)*, we know that the tolerances of V_{in}、 V_{set}、 R have an influence on memristance precision. In this paper, we only discuss the tolerance of V_{set} which is a parameter of memristor intrinsic characteristics. Considering memristor V_{set} has 10% tolerance and assuming the ideal value of V_{set} is V_{set0}, we discuss two conditions:

Condition 1: memristor V_{set} has +10% tolerance,

$$V_{set} = V_{set0}(1 + 10\%) = 1.1V_{set0} \tag{10}$$

Condition 2: memristor V_{set} has -10% tolerance,

$$V_{set} = V_{set0}(1 - 10\%) = 0.9V_{set0} \tag{11}$$

corresponding to bule and red lines in Figure 9., namely, $V_{set} = 2.2V$ and $V_{set} = 1.8V$ based on $V_{set0} = 2.0V$.

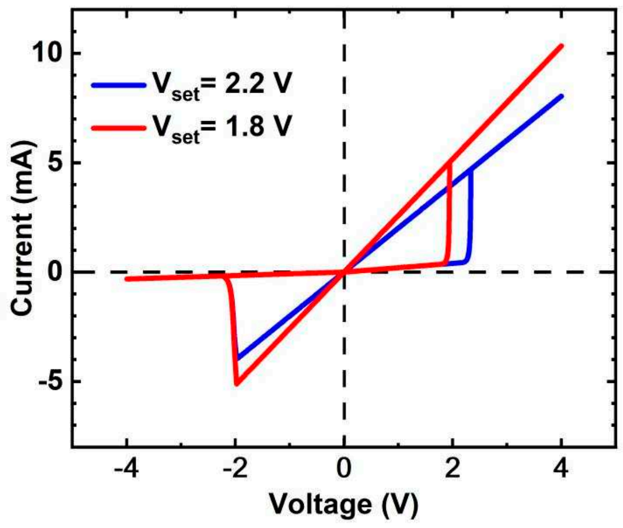

Figure 9. Simulation *I-V* curve of the memristor. The red *I-V* curve is the memristor with the transition voltage of $V_{set} = 1.8V$. The blue *I-V* curve is the memristor with the transition voltage of $V_{set} = 2.2V$.

4.1. The Impact of V_{set} Tolerance

For analysis of the impact of V_{set}, we also consider memristor V_{set} has 10% tolerance and assume the ideal value of V_{set} is V_{set0}, and R_0 is the aim value. We discuss two conditions:

Condition 1: memristor V_{set} has +10% tolerance

When $V_{in} = 2V_{set0}, R = R_0, V_{set} = V_{set0}(1 + 10\%) = 1.1V_{set0}$, the memristance value is

$$R_M = 1.222R_0 \tag{12}$$

In Figure 10., when the positive threshold of the memristor is $V_{set} = 2.2V$ and the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 000$ ($R_{eq} = 1k\Omega$), the simulation value of memristance is $R_M = 1.2224k\Omega$; when the positive threshold of the memristor is $V_{set} = 2.2V$ and the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 111$ ($R_{eq} = 8k\Omega$), the simulation value of memristance is $R_M = 9.7921k\Omega$. The simulation value of memristance conforms to the theoretical value.

Condition 2: memristor V_{set0} has -10% tolerance

When $V_{in} = 2V_{set0}, R = R_0, V_{set} = V_{set0}(1 - 10\%) = 0.9V_{set0}$, the memristance value is

$$R_M = 0.818R_0 \tag{13}$$

In Figure 10., when the positive threshold of the memristor is $V_{set} = 1.8V$ and the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 000$ ($R_{eq} = 1k\Omega$), the simulation value of memristance is $R_M = 0.8181k\Omega$; when the positive threshold of the memristor is $V_{set} = 1.8V$ and the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 111$ ($R_{eq} = 8k\Omega$), the simulation value of memristance is $R_M = 6.5496k\Omega$. The simulation value of memristance also conforms to the theoretical value.

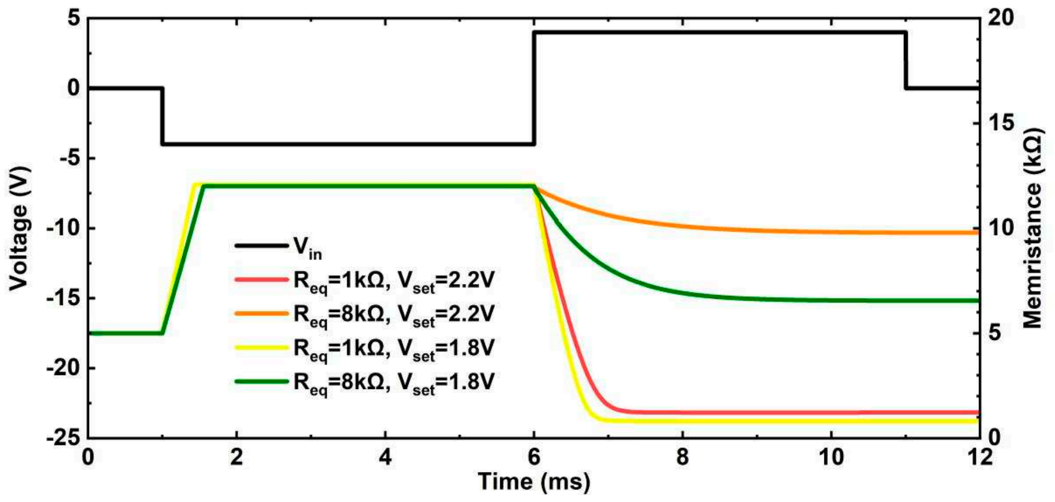

Figure 10. Simulation precision of the circuit. The black line is the input source V_{in}, and the color lines are the memristances based on different transition voltage tolerances as the function of stimulation time.

5. Optimized Circuit

Through the previous analysis, we know the tolerance of V_{set} has an influence on memristance precision. This proposes a solution to optimize the memristance precision by adjusting the values of V_{in} and R. However, it does not need to change the structure of the circuit. For example, when $V_{in} = 3V_{set0}$, $R = 2R_0$, $V_{set} = V_{set0}$ (R_0 is the aim value, V_{set0} is the ideal value), it can still precisely program memristance $R_M = R_0$. In Figure 11., when the positive threshold of the memristor is $V_{set} = 2\text{V}$ and the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 000$ ($R_{eq} = 1\text{k}\Omega$), the simulation value of memristance is $R_M = 1.0001\text{k}\Omega$; when the positive threshold of the memristor is $V_{set} = 2\text{V}$ and the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 111$ ($R_{eq} = 8\text{k}\Omega$), the simulation value of memristance is $R_M = 8.0010\text{k}\Omega$. The memristance simulation results are equal to R_{eq} with an acceptable error. It proves that this solution is effective. Now, considering the tolerance of V_{set}, discuss two conditions:

Condition 1: memristor V_{set} has +10% tolerance

When $V_{in} = 3V_{set0}, R = 2R_0, V_{set} = V_{set0}(1 + 10\%) = 1.1V_{set0}$

$$R_M = 1.158R_0 \tag{14}$$

In Figure 11, when the positive threshold of the memristor is $V_{set} = 2.2\text{V}$ and the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 000$ ($R_{eq} = 1\text{k}\Omega$), the simulation value of memristance is $R_M = 1.1576\text{k}\Omega$; when the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 111$ ($R_{eq} = 8\text{k}\Omega$), the simulation value of memristance is $R_M = 9.2626\text{k}\Omega$; the simulation value of memristance conforms to the theoretical value.

Condition 2: memristor V_{set} has -10% tolerance

When $V_{in} = 3V_{set0}, R = 2R_0, V_{set} = V_{set0}(1-10\%) = 0.9V_{set0}$

$$R_M = 0.857R_0 \tag{15}$$

In Figure 11, when the positive threshold of the memristor is $V_{set} = 1.8\text{V}$ and the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 000$ ($R_{eq} = 1\text{k}\Omega$), the simulation of memristance is $R_M = 0.8569\text{k}\Omega$; when the digital input signals are $\overline{b3}\,\overline{b2}\,\overline{b1} = 111$ ($R_{eq} = 8\text{k}\Omega$), the simulation of memristance is $R_M = 6.8611\text{k}\Omega$; the simulation value of memristance also conforms to the theoretical value. Condition 1 and condition 2 confirm that properly adjusting the values of V_{in} and R can optimize the memristance precision. However, it needs a higher voltage value of source V_{in}, so it should make a tradeoff between memristance precision and input voltage value.

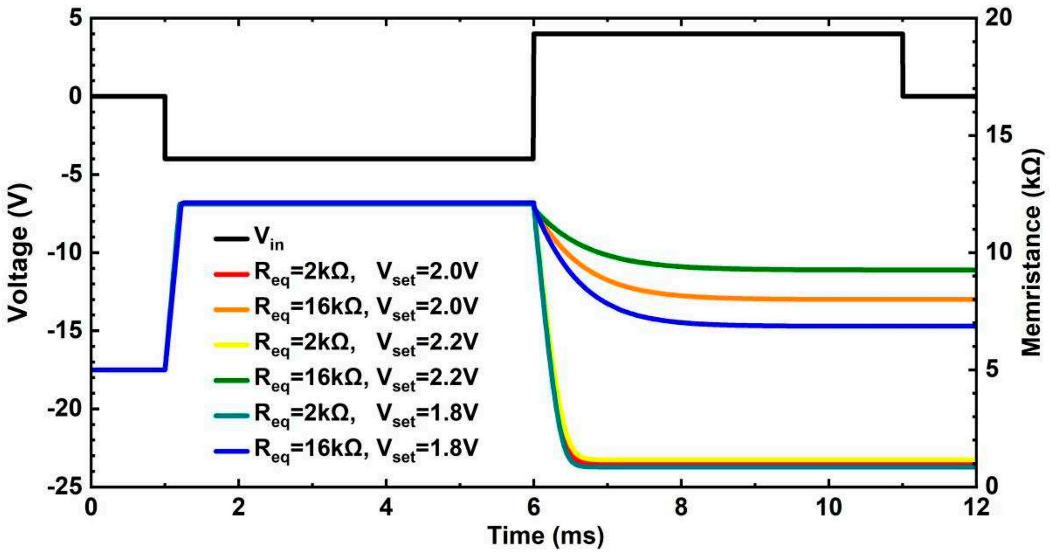

Figure 11. Simulation of the optimized circuit. The black line is the input source V_{in}, and the color lines are the memristances evolution based on optimized circuit.

6. Memristor Crossbar Programming Circuit

Besides the single memristor device, our programming circuit can also be used to control the memristor crossbar precisely. For example, as shown in Figure 12., in an mxn 1T1R memristor crossbar, the row signals are controlled by selectors S_1、S_2、……、S_n, and the column signals are controlled by selectors S_{n+1}、S_{n+2}、……、S_{n+m}. Through selecting specific row signal and column signal, it can program specific memristor in the memristor crossbar precisely. It also takes two steps to program the memristor. In addition, in the former structure, it uses a single source (V_{in}) to program the memristor where a negative pulse is necessary. Here, we use two sources (V_{in1}&V_{in2}) to stimulate the memristor crossbar and these sources are all positive bias.

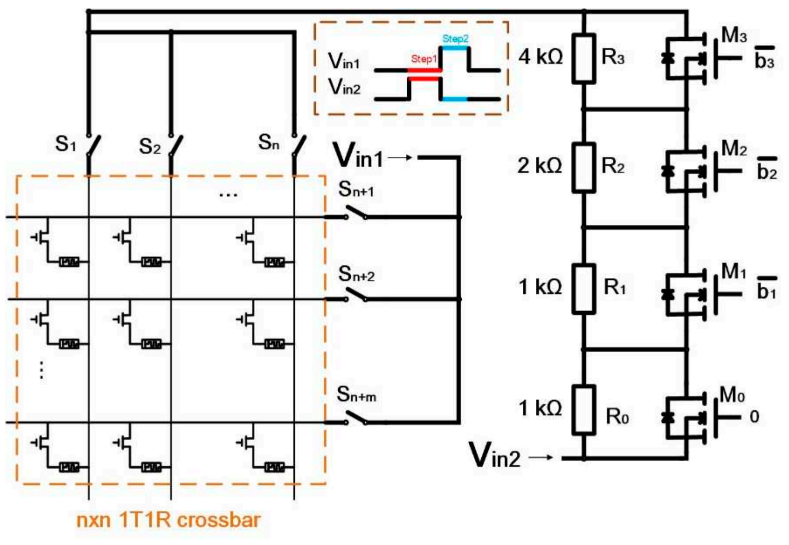

Figure 12. Memristor crossbar programmable circuit. S_1、S_2、……、S_n are the selectors of 1T1R crossbar rows. S_{n+1}、S_{n+2}、……、S_{n+m} are the selectors of 1T1R crossbar columns. V_{in1} and V_{in2} are two input sources.

There are also two steps to program the memristor which is illustrated as follows. Taking an example on a 3x3 1T1R memristor crossbar and the memristor in the second row and second column needs to be updated. In Step1, as in Figure 13.(a), V_{in1} is 0V and V_{in2} is 4V which triggers memristance increases sharply until to R_{off}, the memristor completes RESET process. In Step2, V_{in1} is 4V while the V_{in2} is 0V which makes the memristance of the memristor is programmed to aim value gradually, and the memristor completes the programming process. The programming process for the other memristors is similar.

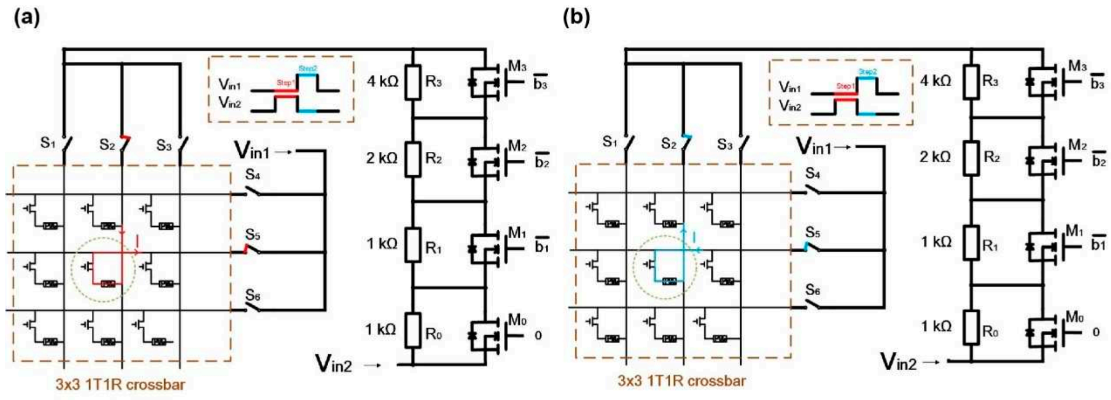

Figure 13. memristor crossbar programmable circuit. The states of selectors S_2 and S_4 are ON; while the states of other selectors are OFF. (a) Step1. In this period, input source V_{in1} is 0V and V_{in2} is 4V. (b) Step2. In this period, input source V_{in2} is 0V and V_{in1} is 4V.

Figure 14. are simulation waveforms, 1-6ms is Step 1, and 6-11ms is Step 2. When the positive threshold of the memristor is $V_{set} = 2\text{V}$ and the digital input signals are $\overline{b3b2b1} = 000$ ($R_{eq} = 1\text{k}\Omega$), the simulation value of memristance is $R_M = 0.9998\text{k}\Omega$; when the digital input signals are $\overline{b3b2b1} = 111$ ($R_{eq} = 8\text{k}\Omega$), the simulation value of memristance is $R_M = 8.008\text{k}\Omega$. The simulation results agree with the aim values and the precision is more than 99%.

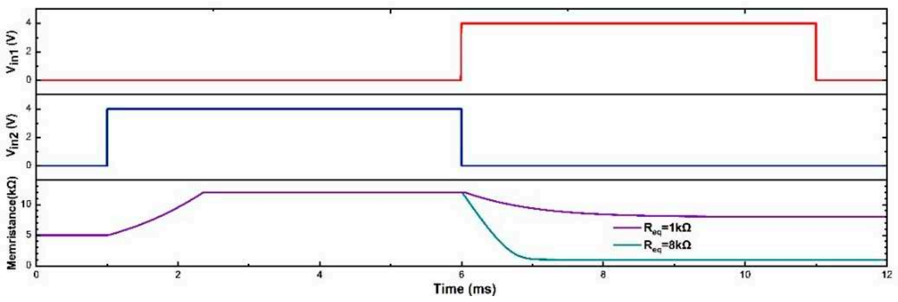

Figure 14. Memristor crossbar programmable circuit simulation. The red line is the source V_{in1}, and the value is 0V during 1-6ms and 4.0V during 6-11ms. The blue line is the source V_{in2}, and the value is 4V during 1-6ms and 0V during 6-11ms. The purple line represents the memristance evolution when the reference resistor is $R_{eq}=8\text{k}\Omega$ and the cyan line represents the memristance evolution when the reference resistor is $R_{eq}=1\text{k}\Omega$.

7. Application

With the advent of the big data era, the scale of information has been explosive growth. And the scale of image data is also increasing explosively. However, in the conventional computing architecture, computation and storage is physically separate. It requires frequent data shuttling among the computation and storage units, which causing significant system consumption and speed loss. Hence, it is difficult to emulate the requirements of information analysis and processing23. . Memristors with nonvolatility and integrated storage-and-computation properties can be used to build intelligent processing systems that are closer to the structure and function of biological brains26. . They are also of great significance for achieving the integrated storage and computation for image data28. .

As Figure 15(a), it is a CNN architecture for image recognition based on memristor crossbar arrays. Inputs are images for recognition, and outputs are the results of recognition. On platform MNSIM29. , we analyzed the computing accuracy of different memristance tolerance. The algorithm uses VGG8 and the dataset uses CIFAR10. The size of input image is 28x28, hence, the network consists of 784 input neurons. In addition, the numbers of convolution layers and pooling layers are 8 and 5, respectively. And 10 output neurons are corresponding to 10 kinds of classification targets. Hardware configuration is Intel Xeon Gold 6248 @2.50GHz CPU. System is Windows Server 2019 and its word length is 64 bits.

From Table 2., taking 10% tolerance of V_{set} into consideration. As mentioned above, the memristance tolerances of the circuit before and after optimization are 22.22% and 15.8%. Hence, we analyzed the computing accuracy of three memristance tolerance 1%, 15.8%, and 22.2%. As Figure 7.(b), the computing accuracy is 93.45% (original accuracy). For computing based on memristor crossbar arrays, when the memristance tolerance is 1%, the computing accuracy is 93.09%, and the computing accuracy decreased to 89.36% and 77.64% with the memristance tolerance of 15.8% and 22.22%, respectively. The latency of the network training directly by traditional computer is about 7.47s, while this value decreased to 18.78ms based on memristor crossbar arrays, and the time efficiency improved more than 396 times.

Table 2. Memristance precision comparison.

	$V_{set}=V_{set0}$	$V_{set}=1.1V_{set0}$	$V_{set}=0.9V_{set0}$
$V_{in}=2V_{set0}, R=R_0$	0	+22.2%	-18.2%
$V_{in}=3V_{set0}, R=2R_0$	0	+15.8%	-14.3%
Precision optimized	0	6.4%	3.9%

Table 3. Simulation result of memristance.

	$V_{set}=V_{set0}$		$V_{set}=1.1V_{set0}$		$V_{set}=0.9V_{set0}$	
	$R_0=1\text{k}\Omega$	$R_0=8\text{k}\Omega$	$R_0=1\text{k}\Omega$	$R_0=8\text{k}\Omega$	$R_0=1\text{k}\Omega$	$R_0=8\text{k}\Omega$

$V_{in} = 2V_{set0}$, $R = R_0$	1.0001kΩ	8.0087kΩ	1.2224kΩ	9.7921kΩ	0.8181kΩ	6.5496kΩ
$V_{in} = 3V_{set0}$, $R = 2R_0$	1.0001kΩ	8.0010kΩ	1.1576kΩ	9.2626kΩ	0.8569kΩ	6.8611kΩ

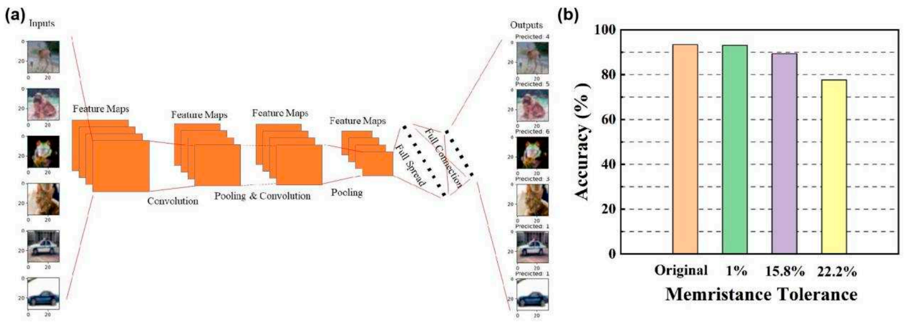

Figure 15. (a) A CNN architecture for images recognition; (b) Computing accuracy of the different memristance tolerance.

8. Conclusion

In this paper, a novel memristor programming circuit with a reference resistor structure is proposed which can achieve precise memristance, and the function of the circuit has been fully verified by simulation. Just as the memristance adjustment is sensitive to the memristor V_{set}, this paper analyzed the precision of the memristance under the memristor V_{set} has ±10% tolerance, and proposed a solution to improve the memristance precision at the cost of increasing input voltage value without changing the circuit structure. In order to complete the circuit design, it should make a tradeoff between the memristance precision and the input voltage value. In addition, this paper demonstrates that the circuit can be applied to the memristor crossbar and the simulation results of the circuit verify the programming function on the memristor crossbar. In the end, this paper analyzed the computing accuracy of different memristance tolerance in the CNN image recognition based on memristor crossbar arrays.

Author Contributions: Direction of the research , Shisheng Xiong; methodology , Shengtao Tu; circuit design, Shengtao Tu; simulation of the circuit, Shengtao Tu; software, Jinyu Li. writing—original draft preparation, Shengtao Tu; writing—review and editing, Shengtao Tu, Shisheng Xiong, Yanyun Ren, and Qin Jiang; All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by National Natural Science Foundation of China (Nos. U20A20227).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chua, L. Memristor-the missing circuit element. *IEEE Trans. Circuit Theory* **1971**, *18*, 507–519.
2. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S. The missing memristor found. *Nature* **2008**, *453*, 80–83.
3. Jiang, Q.; Ren, Y.Y.; Cui, Z.J.; Li, Z.L.; Hu, L.G.; Guo, R.Q.; Duan, S.K.; Xie, S.X.; Zhou, G.D.; Xiong, S.S. CsPbBr3 Perovskite Quantum Dots Embedded in Polystyrene-poly2-vinyl Pyridine Copolymer for Robust and Light-Tunable Memristors. *ACS Applied Nano Materials* **2023**, *6*, 8655-8667.
4. Yang, X.; Taylor, B.; Wu, A.; Chen, Y.; Chua, L.O. Research progress on memristor: From synapses to computing systems. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **2022**, *69*, 1845–1857.
5. Ji, X.; Dong, Z.; Lai, C.S.; Qi, D. A brain-inspired in-memory computing system for neuronal communication via memristive circuits. *IEEE Commun. Mag.* **2022**, *60*, 100–106.
6. Zhong, Y.; Tang, J.; Li, X.; Liang, X.; Liu, Z.; Li, Y.; Xi, Y.; Yao, P.; Hao, Z.; Gao, B.; et al. A memristor-based analogue reservoir computing system for real- time and power-efficient signal processing. *Nat. Electron.* **2022**, *5*, 672–681.

7. Yoo, J.; Song, H.; Lee, H.; Lim, S.; Kim, S.; Heo, K.; Bae, H. Recent Research for HZO-Based Ferroelectric Memory towards In-Memory Computing Applications. *Electronics* **2023**, *12*, 2297.
8. Dong, Z.; Ji, X.; Zhou, G.; Gao, M.; Qi, D. Multimodal neuromorphic sensory-processing system with memristor circuits for smart home applications. *IEEE Trans. Ind. Appl.* **2022**.
9. Lai, C.S.; Dong, Z.; Qi, D. Memristive Devices and Systems: Modeling, Properties and Applications. *Electronics* **2023**, *12*, 765.
10. Pershin, Y.V.; Ventra, M.D. Practical Approach to Programmable Analog Circuits with Memristors. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **2010**, *57*, 1857-1864.
11. Berdan, R.; Prodromakis, T.; Toumazou, C. High Precision Analogue Memristor State Tuning. *Electronics Letters* **2012**, *48*, 1105-1107.
12. Merced-Grafals, E.J.; Dávila; Noraica; Ge, N.; Williams, R. S.; Strachan, J. P. Repeatable, Accurate, and High Speed Multi-Level Programming of Memristor 1T1R Arrays for Power Efficient Analog Computing Applications. *Nanotechnology* **2016**, *27*, 365202.
13. Kim, K.; Yang, J.; Strachan, J. Voltage Divider Effect for the Improvement of Variability and Endurance of TaO_x Memristor. *Sci Rep* **2016**, *6*, 20085.
14. Vourkas, I.; Gómez, J.; Abusleme, Á.; Vasileiadis, N.; Sirakoulis G.C.; Rubio, A. Exploring the Voltage Divider Approach for Accurate Memristor State Tuning. *IEEE 8th Latin American Symposium on Circuits & Systems (LASCAS), Bariloche, Argentina* **2017**, 1-4.
15. Gomez, J.; Vourkas, I.; Abusleme, A.; Sirakoulis, G.C.; Rubio, A. Voltage Divider for Self-Limited Analog State Programing of Memristors. *IEEE Transactions on Circuits and Systems II: Express Briefs* **2019**, *99*, 1-1.
16. Olumodeji, O.A.; Gottardi, M. A Pulse-Based Memristor Programming Circuit *IEEE International Symposium on Circuits & Systems* **2017**, 1-4.
17. Mokhtar, S.; Wan, F.; Kadiran, K.A.; Rifin, R.; Omar, M. Write and Read Circuit for Memristor Analog Resistance Switching. *IEEE 8th Control and System Graduate Research Colloquium (ICSGRC)* **2017**.
18. Tarkhan, M.; Maymandi-Nejad, M.; Klidbary, S.H.; Shouraki, S.B. A Bridge Technique for Memristor State Programming. *International Journal of Electronics: Theoretical & Experimental* **2020**, *107*, 1015-1030.
19. Cirera, A.; Fernandez, C.; Vourkas, I.; Rubio, A. Exploring Different Circuit-level Approaches to the Forming of Resistive Random Access Memories. *11th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany* **2022**, 1-4.
20. Lu, W.; Bao, N.; Zheng, T.; Zhang, X.; Song, Y. Memristor-Based Read/Write Circuit with Stable Continuous Read Operation. *Electronics* **2022**, *11*, 2018.
21. Randrianantenaina, J.L.; Baran, A.Y.; Korkmaz, N. Functional Emulator Designs for a Memristor Model with Programmable Analog and Digital Platforms. *Journal of Computational Electronics* **2023**, *22*, 519–530.
22. Pershin, Y.V.; Ventra, M.D. SPICE Model of Memristive Devices with Threshold. *Radioengineering* **2013**, *22*, 485-489.
23. Zidan, M.A.; Strachan, J.P.; Lu, W.D. The future of electronics based on memristive systems. *Nat. Electron.* **2018**, *1*, 22–29.
24. Wang, Z.; Joshi, S.; Savel'ev, S.E.; Jiang, H.; Midya, R.; Lin, P.; Hu, M.; Ge, N.; Strachan, J.P.; Li, Z.; et al. Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. *Nat. Mater.* **2017**, *16*, 101–108.
25. Dong, Z.; Sing Lai, C.; Zhang, Z.; Qi, D.; Gao, M.; Duan, S. Neuromorphic extreme learning machines with bimodal memristive synapses. *Neurocomputing* **2021**, *453*, 38–49.
26. Schuman, C.D.; Potok, T.E.; Patton, R.M.; Birdwell, J.D.; Dean, M.E.; Rose, G.S.; Plank, J.S. A survey of neuromorphic computing and neural networks in hardware. *arXiv* **2017**, arXiv:1705.06963.
27. Davies, M.; Srinivasa, N.; Lin, T.H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **2018**, *38*, 82–99.
28. Ji, X.; Dong, Z.; Zhou, G.; Lai, C.S.; Yan, Y.; Qi, D. Memristive System Based Image Processing Technology: A Review and Perspective. *Electronics* **2021**, *10*, 3176.
29. Xia, L. MNSIM: Simulation Platform for Memristor-Based Neuromorphic Computing System. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **2018**, *37*, 1009-1022.