sensors

MDPI

Article

Convolution Kernel Operations on a Two-Dimensional Spin Memristor Cross Array

Saike Zhu [1,2], Lidan Wang [1,2,*], Zhekang Dong [3] and Shukai Duan [1,2]

1 School of Electronic Information Engineering, Southwest University, Chongqing 400715, China; saikezhu@email.swu.edu.cn (S.Z.); duansk@swu.edu.cn (S.D.)
2 Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing 400715, China
3 School of Electronic Information, Hangzhou Dianzi University, Hangzhou 310018, China; englishp@hdu.edu.cn
* Correspondence: ldwang@swu.edu.cn

Received: 15 October 2020; Accepted: 29 October 2020; Published: 31 October 2020

Abstract: In recent years, convolution operations often consume a lot of time and energy in deep learning algorithms, and convolution is usually used to remove noise or extract the edges of an image. However, under data-intensive conditions, frequent operations of the above algorithms will cause a significant memory/communication burden to the computing system. This paper proposes a circuit based on spin memristor cross array to solve the problems mentioned above. First, a logic switch based on spin memristors is proposed, which realizes the control of the memristor cross array. Secondly, a new type of spin memristor cross array and peripheral circuits is proposed, which realizes the multiplication and addition operation in the convolution operation and significantly alleviates the computational memory bottleneck. At last, the color image filtering and edge extraction simulation are carried out. By calculating the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) of the image result, the processing effects of different operators are compared, and the correctness of the circuit is verified.

Keywords: spin memristor; mask operation; memristor switch; memristor crossbar; image processing

1. Introduction

In recent years, in emerging data-intensive applications such as deep convolutional neural networks (DCNN) [1,2], image preprocessing tends to consume a lot of time and energy. The convolution processing operation is a widely applicable and far-reaching algorithm in the field of image processing. When convolution operations on images involve multiplication and addition operations, there will have problems of relatively large overhead, high real-time requirement, strong concurrency and frequent data exchange between memory and processor. The existing processing architecture has insufficient memory bandwidth and processing performance, resulting in a "memory wall" effect. Because the technological development of CMOS technology has reached its limit [3–5], the focus has been shifted from seeking faster processors to alleviating "memory bottlenecks". Although multicore architecture [6] and near-data structure [7,8] have been tried to improve convolution computing performance, they have not jumped out of the storage and computing separation architecture under the Von Neumann system, and there are still problems such as low energy/area efficiency, expensive hardware cost [9–11], etc. Memristor has the characteristics of integration of storage and calculation, so it has a wide range of application prospects in the field of in-memory computing.

Since 1971, Professor Chua predicted the existence of the fourth basic circuit device—memristor [12], and scientists have been searching for the missing memristor for 37 years.

Until 2008, HP laboratory [13] achieved the physical entity components of the memristor. Since then, new memristor devices and memristor models have been published [14,15]. The size of the memristor has reached the nanometer size, with good performance, storage and calculation integration, low energy consumption, short time, and good real-time performance in switching states. The memristance is controlled by electrical signals and has good non-volatile characteristics. It is compatible with CMOS technology in the production process. Therefore, the memristor has wide application prospects in the fields of image compression and filtering [16], image recognition [17], perceptron network [18], logic gate circuit [19,20], pulse neural network [21,22], non-volatile RAM [23], neural network synapse [24–26], establishment of chaotic circuits [27,28], and reconfigurable analog circuits [29], etc.

The spin memristor is a memristor model based on the magnetic domain wall movement mechanism. The resistance value will change only when the real-time current density is greater than or equal to the current density threshold. Therefore, the spin memristor has the characteristics of strong anti-interference ability and multi-level stability. Compared with memristors based on ion transport, it has better controllability [30]. Under the action of external excitation, the read and write speed of the spin memristor model can reach the nanosecond level. At the same time, the spin memristor has ultrafast spin dynamics [30–32] characteristics and good linearity, the smallest size, and compatibility with CMOS technology [7], so the spin memristor arrays are conducive to large-scale memory and high-speed logic operations. The spin memristor is a memristor model with a current threshold, so it has a more significant switching characteristic. The memristance value can quickly switch between high resistance and low resistance, and the constant voltage effect can be derived by mathematical deduction. The specific expression of the time required for the resistance change of the spin memristor under constant voltage can be obtained by mathematical deduction. Thus, the application time of external excitation can be effectively controlled and the unnecessary energy loss in memristive circuit can be reduced. Therefore, it can be designed as a flexible circuit switch.

Moreover, [33] proposed a memristor cross array, which greatly reduces the power consumption of the cross array and reduces the area of the cross array, but no specific application is given. A self-renewing mask circuit is proposed in [34], which uses a multibit memristor cross array to achieve convolution operation and realizes mean filtering and edge extraction. However, a 3×3 convolution kernel is used to perform sliding convolution on the entire image. During operation, each time the 3×3 image block is multiplied and added, the entire circuit needs to be used, which is very expensive. Moreover, when [35] used a 12×12 cross array, the 2D matrix is reduced to a 1D column vector, and multiple 2D convolution kernels are allowed to be read in parallel. The convolution operation of the convolution kernel is realized, but the peripheral circuit and control module are not designed.

This paper first proposes a logic switch circuit (MS) based on spin memristors, which realizes the AND gate operation to control the memristors cross array. By controlling the logic switch circuit, different numbers of convolution operators can be stored in the memristors cross array in the form of conductance. Since the cost of convolution operation mainly lies in the calculation of multiplication and addition, the pixels of the three channels of R, G, and B of the color image can be processed and converted into voltages, which input from the bottom to the memristor cross array to realize multiplication and addition operations, and realize different convolution operations. The memory bottleneck of computing is alleviated, and real-time parallel processing is realized. This paper verifies the correctness of the circuit through two image processing simulations. In the first simulation, the circuit is used to achieve the denoising and sharpening of the color image. The filtering effects of five image denoising operators, SRMC operator, median filter operator, 3×3 Gaussian filter operator, 5×5 Gaussian filter operator, and image sharpening operator, are compared. The second simulation is an edge detector, which simulates the extraction results of five image edge extraction operators: Prewitt operator, Sobel operator, Kirsch operator, Robert operator, and Laplacian operator. Two new types of operators based on Prewitt operator and Sobel operator are proposed. The image edge extraction

effect has been significancly improved. By calculating peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), the filtering and edge extraction effects of the above operators are compared.

The rest of the work in the paper is arranged as follows: Section 2 introduces the spin memristor model and proposes a logic switch based on the spin memristor. In Section 3, we will introduce the spin memristor cross array and its peripheral circuits that implement the convolution operation. In Section 4, the MATLAB simulation results of the circuit (mentioned in Section 3) in the image processing application are given, and the performances are analyzed and discussed respectively by adopting different filter operator and edge extraction operator. Finally, Section 5 summarizes the research content of this paper.

2. Logic Switch Based on Spin Memristor

The convolutional cross-array circuit proposed in this paper includes three significant modules—memristor cross-array, row/column address selector containing logic switches, and weighted average filter. The memristor cross-aray can store information in the form of conductance. By controlling the row/column address selector, when a voltage is applied to the corresponding column of the memristor cross-array, the current will carry the memristor information, and added together by an operational amplifier. The average filter can filter out polluted pixels in advance. This implementation has the advantages of parallelism, high efficiency, low complexity, fast speed, and easy control (low control voltage). This section will introduce the spin memristor model and the specific principles of spin memristor-based logic switches.

2.1. Introduction to Spin Memristors

The spin memristor is a device controlled by electric charge. Its structure diagram and equivalent circuit diagram are shown in Figure 1. The spin memristor is composed of a long spin-valve bar, which contains two layers of ferromagnets: The upper layer is the free layer, and the lower layer is the reference layer. The magnetic polarity of the reference layer is fixed in the magnetic layer by coupling technology. The free layer is divided into two by the magnetic domain wall. The two sections have opposite polarities. The resistance of each section is determined by the relative magnetization direction of the reference layer and free layer. When the magnetization directions of the two layers are opposite (parallel), the resistance of the spin memristor reaches the maximum (minimum).

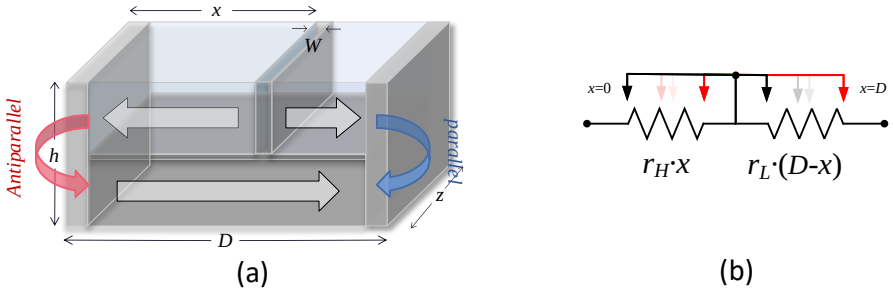


Figure 1. Spin memristor. (**a**) Structure diagram. (**b**) Equivalent circuit diagram.

The length, width, and height of the memristor in Figure 1a are represented by D, h, and z, respectively. W is the thickness of the magnetic domain wall. When power is applied to both ends of the spin memristor, the magnetic domain wall will move in the free layer, the total length of the two magnetization directions of the layer will change, leading to the change of the memristance value. Here, r_L and r_H respectively represent the high and low resistance values of the spin memristor, x represents the distance moved by the magnetic domain wall. Regardless of the domain wall width W, the equivalent circuit diagram of the memristor can be approximated by Figure 1b, and the resistance value of the memristor [15] is:

$$M(x) = r_H \cdot x + r_L (D - x) \tag{1}$$

The relationship between the moving speed v of the domain wall and the current density J is expressed as [15]:

$$v = \frac{dx}{dt} = \begin{cases} \Gamma_v \cdot J = \frac{\Gamma_v}{h_s \cdot z_s} \cdot \frac{dq}{dt}, & J \geq J_{cr} \\ 0, & J < J_{cr} \end{cases} \tag{2}$$

The relationship between x and the amount of charge passing through the memristor is:

$$x(t) = x_0 + \frac{\Gamma_v}{h \cdot z} \cdot q(t) \tag{3}$$

Equation (1) shows that the spintronic memristor has good linearity and can meet the requirements of multibit data storage. Here,Γ_v is the proportional coefficient, which is related to the structure of the device and the properties of the material. Besides, the adjustment of the memristance value $M_{(x)}$ is limited by the current density threshold. In other words, when the current density J is less than the critical current density J_{cr}, the resistance of the spin memristor is a constant. As long as the current density J is less than the critical current density, the state of the spintronic memristor will remain unchanged no matter how long the sensing voltage is maintained. When the current density J is higher than J_{cr}, the memristance value begins to change. The equation for calculating the current threshold J_{cr} of the device is as follows [15]:

$$J_{cr} = \frac{\alpha \gamma H_P e M_s}{P u_B} \sqrt{\frac{2A}{M_s H_k}} \tag{4}$$

where P represents the magnetic susceptibility of the material, M_s represents the saturation magnetization, α and γ represent the damping parameter and gyromagnetic ratio of the memristor, respectively. H_p and H_k represent the hard anisotropy and easy anisotropy of the magnetic material, respectively. A represents the exchange parameter, u_B is the Bohr magneton constant, e is the elementary charge. The expression of the current density J of the spin memristor device is [15]:

$$J = \frac{V}{M(x) \cdot h \cdot z} \tag{5}$$

As shown in Figure 2, the state variable x of the spin memristor model will gradually increase under the action of a positive voltage until the threshold condition is met or the maximum value of the state variable x is reached. The state variable x of the spin memristor model will gradually decrease under the action of negative voltage until the threshold condition is met or the minimum value of the state variable x is approached.

Figure 3 shows the resistance change curve of the spin memristor and its external excitation voltage curve. V is the voltage applied to the memristor, and $M_{(x)}$ represents the resistance value of the spin memristor when the magnetic domain wall moves by the distance x. Specifically, when the external excitation voltage is V_1 (red line), in the time domain [50, 100 ns], the external excitation voltage is positive and satisfied $V_1 / (r_{\mathrm{H}} \cdot h \cdot z) \geq J_{\mathrm{cr}}$, and the memristance value gradually increases to its maximum value; correspondingly, in the time domain [100 ns, 150 ns], when the external excitation voltage is negative and satisfied $|V_1 / (r_{\mathrm{H}} \cdot h \cdot z)| \geq J_{\mathrm{cr}}$, the memristance value gradually decreases to its minimum value. When the external excitation voltage is V_2 (blue line), in the time domain [50, 100 ns], the external excitation voltage is positive and satisfies the double inequality relationship $V_2 / (r_H \cdot h \cdot z) > J_{cr} > V_2 / (r_L \cdot h \cdot z)$, the memristance value gradually increases and the real-time current density decreases. If the real-time current density J is equal to the threshold current density J_{cr}, the memristor will remain unchanged. At this time, in the time domain [100 ns, 150 ns], only the polarity of the excitation voltage is changed, the resistance value of the memristor will not change. Appropriately increasing the amplitude of the excitation voltage to V_3 (green line), so that

the external excitation voltage is negative and satisfied $|V_3/(r_H \cdot h \cdot z)| \geq J_{cr}$, the memristor resistance will decrease, and the real-time current density will increase and is always larger than the threshold current density J_{cr}, the memristance value eventually decreases to its minimum value. Similarly, the time required for the above-mentioned spin memristor resistance change can be derived from Equations (11) and (12).

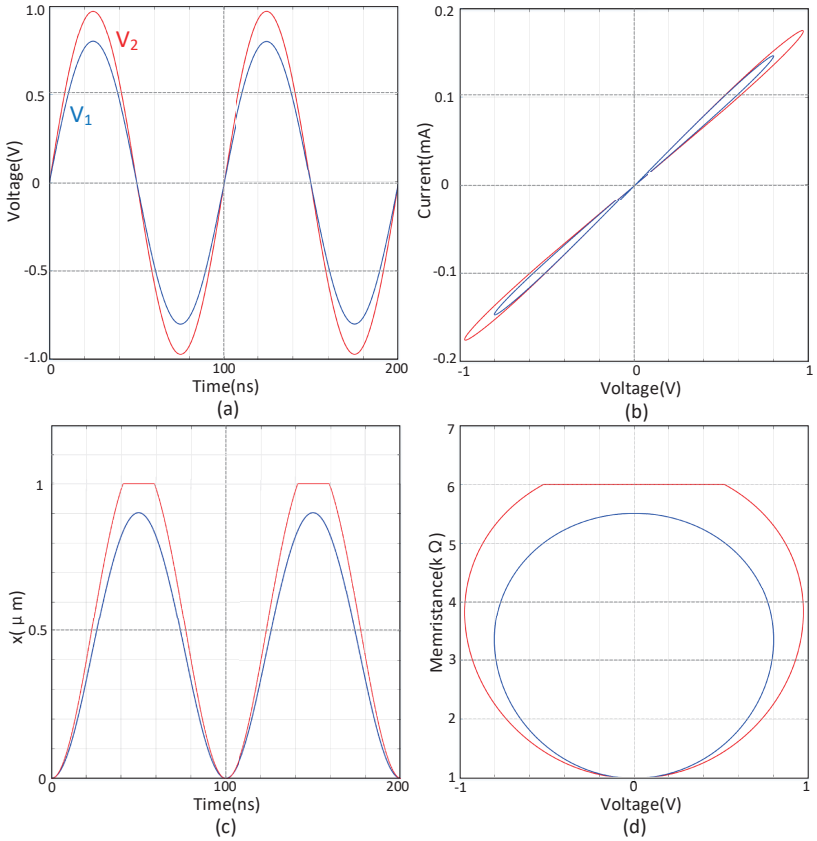


Figure 2. Simulation results of spin memristor ($V_1 < V_2$). (**a**) Time-Voltage curve, (**b**) Voltage-Current curve, (**c**) Time-Magnetic domain wall moving distance curve, (**d**) Voltage-Memristance curve.

The spin memristive damping parameter used in this paper is 0.002, and the current density threshold J_{cr} is approximately equal to 5.74×10^{12}, and the critical current can be calculated as $I_{cr} = J_{cr} \times h \times z = 4.018 \times 10^{-4}$A. After determining the structure and material of the device, the length D and width z of the spintronic memristor are the main factors for adjusting the high and low resistance of spin memristors. Besides, to reduce the localization of changes, the high-impedance resistance value of each cross-point memristor is set to 8 KΩ, and it is connected in series with a fixed resistance of 92 KΩ and a diode. The diode causes the current cross array to flow unidirectionally to prevent the sneak path current from affecting the output result. Adding a fixed resistor can prevent the memristance value from changing too small. In the following content, we will use "memristor" to refer to the memristor, resistor, and diode on the same contact, because the resistance part does not affect the training of the current control memristor. When a constant current exceeding the critical value is applied, the memristance value will decrease. Note that the adjustment in this study does not exceed one ns. By changing the critical current density J_{cr}, training current $I_{tr}(t)$, and speed coefficient v, the speed v of memristive decay can be adjusted.

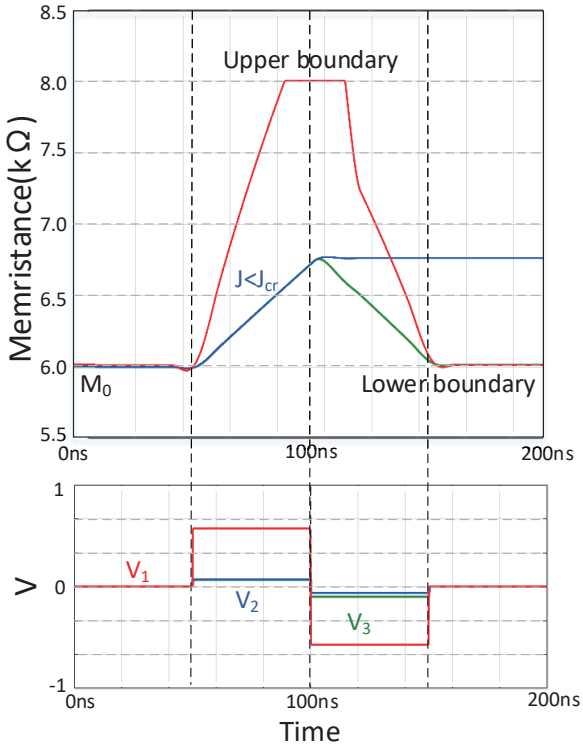


Figure 3. The change rule of the resistance value of the spin memristor model under the action of a voltage source of a fixed amplitude.

2.2. Memristor Switch (MS) Based on Magnetic Flux Control Spin Memristor

The resistance of the spin memristor can be described as [15],

$$M(x) = r_L \cdot D + (r_H - r_L) \cdot x \tag{6}$$

where M is the resistance value of the spin memristor, r_H and r_L are the high resistance and low resistance of the spin memristor, respectively, and D represents the length of the spin memristor. Spin memristor models all have unique threshold characteristics. That is, when the real-time current density J of the spin memristor is less than the threshold current density J_{cr}, it appears as a constant value resistance. On the contrary, when the real-time current density J of the spin memristor is greater than or equal to the threshold current density J_{cr}, the domain wall shifts and changes the resistance of the spin memristor. In this case ($J \geq J_{cr}$), perform differential operations on both sides of Equation (6), we can obtain:

$$\frac{dM}{dt} = (r_H - r_L) \cdot \Gamma_v \cdot J = \frac{(r_H - r_L) \cdot \Gamma_v}{h \cdot z} \cdot \frac{V}{M} \tag{7}$$

where Γ_v is the proportional coefficient, J represents the current density, h and z represent the width and height of the spin memristor, respectively, and V is the voltage applied on the spin memristor.

Then, integrate both sides of Equation (7) at the same time, to obtain the expression of the resistance value of the spin memristor controlled by the magnetic flux as follows:

$$M(\varphi) = \begin{cases} r_{\mathrm{H}}, & \varphi > \varphi_{\mathrm{th2}} \\ \sqrt{M_0^2 + 2B \cdot \varphi}, \varphi_{\mathrm{th1}} \leq \varphi, & \varphi_{\mathrm{th1}} \leq \varphi \leq \varphi_{\mathrm{th2}} \\ r_{\mathrm{L}}, & \varphi < \varphi_{\mathrm{th1}} \end{cases} \tag{8}$$

where

$$\begin{cases} \varphi_{\text{th1}} = \frac{r_{\text{L}}^2 - M_0^2}{2B} \\ \varphi_{\text{th2}} = \frac{r_{\text{H}}^2 - M_0^2}{2B} \end{cases} \tag{9}$$

where M_0 is the initial resistance value of the spin memristor, φ is the magnetic flux flowing through the memristor, and its corresponding threshold φ_{th1} ,φ_{th2} depends on the limit memristance value and the initial memristance value of the spin memristor. The auxiliary variable $B = \Delta r \cdot \Gamma_v / z \cdot h$ is a fixed constant.

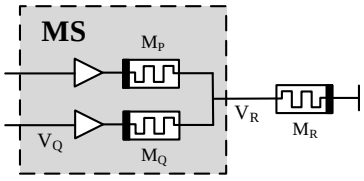
In particular, based on Equation (8), the input voltage V is a constant with a fixed amplitude and the real-time current density of the memristor always satisfies $J \geq J_{cr}$. When the memristance changes from r_H to r_L, the total magnetic flux inside the spin memristor changes as follows:

$$\Delta\varphi = \frac{1}{2B}\left[r_H^2 - r_L^2\right] \tag{10}$$

The proposed AND logic switch (MS) based on the memristor is a simplified form of [36], which consists of two memristors P and Q connected through two positive terminals, as shown in Figure 4a. V_P and V_Q are two input voltages, and V_R is the output. In order to ensure the correctness of AND logic operation, $R_R \gg R_P, R_Q$. The truth table based on the AND operation of the memristor is shown in Figure 4b. The time required to change the memristance from r_L to r_H or from r_H to r_L is assumed to be the same. The initial states of the P and Q memristors are arbitrary. Based on $\Delta\varphi = V \cdot \Delta T$, the MS switching time T_1, the M_R switching time T_2 is

$$T_1 = \frac{1}{2B \cdot V_P}\left|r_H^2 - r_L^2\right| \tag{11}$$

$$T_2 = \frac{1}{2B \cdot V_R}\left|{r'_H}^2 - {r'_L}^2\right| \tag{12}$$



(a)

V_P	V_Q	V_R
0	0	0
1	0	0
0	1	0
1	1	1

(b)

Figure 4. Logic switch based on spin memristor. (**a**) Logical switch based on memristor. (**b**) Truth table for AND operation.

r_H and r_L are the high resistance and low resistance of the memristor P and Q, respectively, and $r_H{}'$ and $r_L{}'$ are the high resistance and low resistance of the memristor M_R, as shown in Figure 4a. Assuming there is no threshold in MS. The output error V_e [36] is

$$V_{\text{e}} \approx \frac{R_{\text{Q}}}{R_{\text{P}} + R_{\text{Q}}} V_H = \frac{r_{\text{L}}}{r_{\text{H}} + r_{\text{L}}} V_H \tag{13}$$

According to the simulation results of the spin memristor logic switch in Figure 5, there are four special cases as follow:

1. $V_P = V_Q = V_H$ = "1" ("1" stands for logic 1, V_H stands for high-level voltage; "0" stands for logic 0, V_L stands for low-level voltage, and $V_L = 0$), the output voltage V_R is

$$V_{\mathrm{R}} \approx \frac{R_{\mathrm{Q}} + R_{\mathrm{P}}}{R_{\mathrm{P}} + R_{\mathrm{Q}}} V_{\mathrm{P}} = V_{\mathrm{P}} \equiv V_{\mathrm{H}} \tag{14}$$

Since there is a positive voltage across M_R, its memristive is reduced. After time T_2 , $R_R = R_{on}$, the logic value stored in M_R changes to logic 1.

2. $V_P = V_H$ = "1", $V_Q = V_L$ = "0", the voltage across M_P is negative, and the voltage across M_Q is positive. After time T_1, $R_P = r_H$,$R_P = r_L (R_{on} \ll r_H)$, the output voltage V_R is

$$V_{\mathrm{R}} \approx \frac{R_{\mathrm{Q}}}{R_{\mathrm{P}} + R_{\mathrm{Q}}} V_{\mathrm{P}} = \frac{r_L}{r_H + r_L} V_{\mathrm{P}} \approx 0 \tag{15}$$

The logical value is stored in M_R to retain logic 0.

3. $V_P = V_L$ = "0", $V_Q = V_H$ = "1", the voltage across M_P is positive, and the voltage across M_Q is negative. After time T_1, $R_P = r_L$, $R_Q = r_H$, the output voltage V_R is

$$V_{\mathrm{R}} \approx \frac{R_{\mathrm{P}}}{R_{\mathrm{P}} + R_{\mathrm{Q}}} V_{\mathrm{Q}} = \frac{r_L}{r_L + r_H} V_{\mathrm{Q}} \approx 0 \tag{16}$$

The logical value is stored in M_R to retain logic 0.

4. $V_P = V_Q = V_L$ = "0". The output voltage $V_R = 0$, so the logic value is stored in M_R to retain logic 0. Therefore, the total time required for a complete logic switch operation is

$$T = T_1 + T_2 = \frac{1}{2B \cdot V_P} \left| r_H^2 - r_L^2 \right| + \frac{1}{2B \cdot V_R} \left| {r'_H}^2 - {r'_L}^2 \right| \tag{17}$$

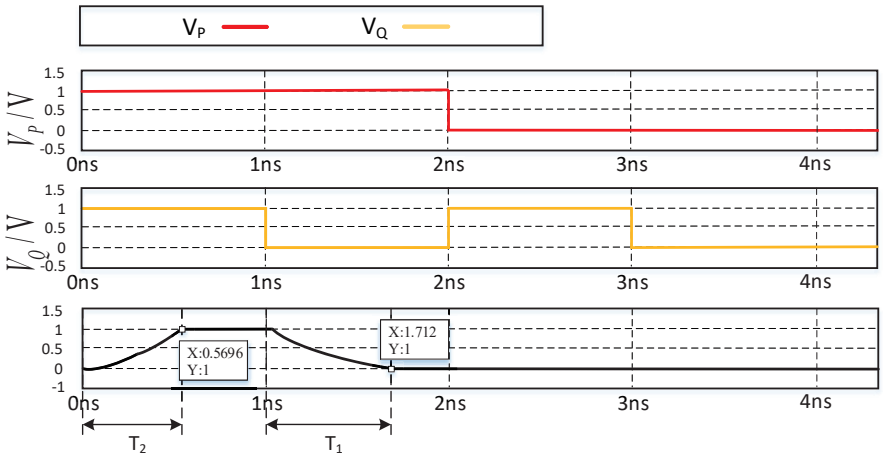
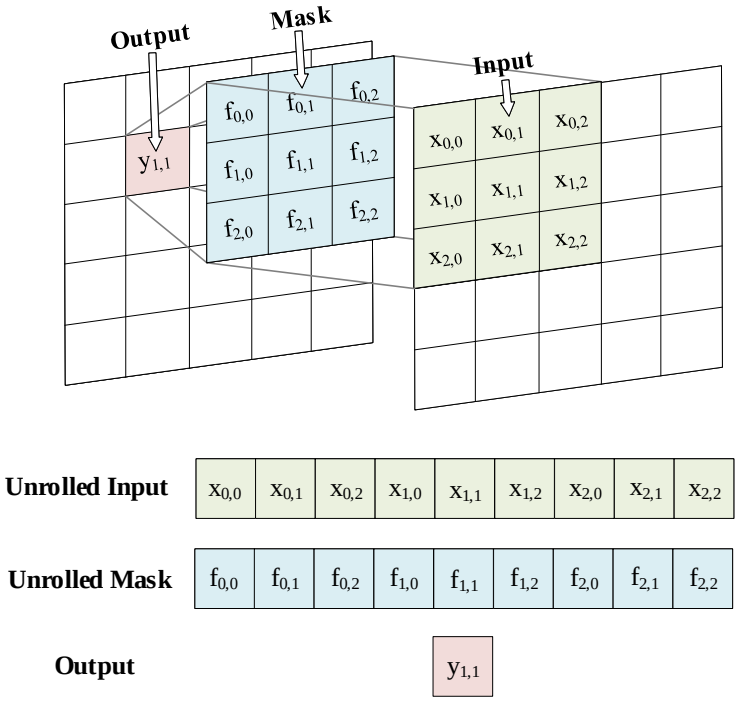


Figure 5. Simulation results of logic switches based on spin memristor.

3. Spin Memristor Cross-Array Circuit for Realizing Convolution Operation

The convolution kernel (also known as the filter) is usually a 2D matrix. To implement the convolution kernel on the memristor array, the 2D convolution kernel matrix is expanded and its dimensionality is reduced to 1D column vector. The paper uses the N × N convolution kernel matrix; here, we choose a convolution kernel with an odd number of N because if the filter size is even, the size of the input and output cannot be guaranteed unchanged. As shown in Figure 6, the convolution operation is performed on each pixel of the input image and involves three consecutive steps. When the convolution kernel is superimposed on the input image in this way, the operation starts. At first, the image to be processed is padded, and the center pixel of the convolution kernel is aligned with the single-pixel which is convolved in the input image. Then, multiply each pixel value in the input image by the corresponding value in the kernel. In the third step, the sum of the products of the second step

is calculated, and the sum becomes the pixel value in the output. To convolve the entire image, it must be moved and received repeatedly to scan the input image pixel by pixel.



$$y_{m,n} = x_{m,n} \times f_{m,n} = \sum_{j} \sum_{i} x_{i,j} f_{i,j}$$

Figure 6. 2D matrix convolution.

3.1. Cross Array Circuit Based on Spin Memristor

Assume that a system is in discrete iteration V_k of the input, and the index is $V_k = 1, 2, \ldots, V_m$. During each iteration k, the system will receive a pair of two vectors M and N of size: the input image matrix $V_I^k \in R^M$ and the output image matrix $V_O^k \in R^N$. Assuming that the F convolution kernel array is an adjustable $N \times M$ matrix, and considering the estimator,

$$V_{out,j}^{(k)} = \sum_{i=1}^{M} F_{ji}^{(k)} V_{in,i}^{(k)} \tag{18}$$

where $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$.

Figure 7 shows the memristor cross array used for convolution operations, which consists of a single crossbar array of $M_{(g,k)}$ and a $1/R_B$ constant term circuit. By expanding the pixels of an input image and converting them into voltages to be applied to rows, the currents read from multiple columns in an array can be calculated in parallel with the convolution results, which significantly accelerates the calculation speed. Here, $g_{i,j}$ is the memristor conductance at the intersection between the jth row and the kth column. $V_{in,j}$ is the input voltage applied to the jth column. $V_{C,K}$ is the row line voltage on the kth row. The row lines $V_{C,F}$ are connected to all the applied input voltages from $V_{in,1}$ to $V_{in,m}$ through R_B. In Figure 7, $V_{C,F}$, and negative feedback resistor R_1 enter G_F together. The latter constitutes an inverting amplifier. The output voltage of G_F is V_F, which is connected to all row lines from $V_{C,1}$ to $V_{C,n}$ through R_1. By applying Kirchhoff's current law to the row lines $V_{C,F}$, we can calculate V_F [33] as:

$$V_F = -\sum_{j=1}^{m} \frac{R_1}{R_B} V_{IN,j} \tag{19}$$

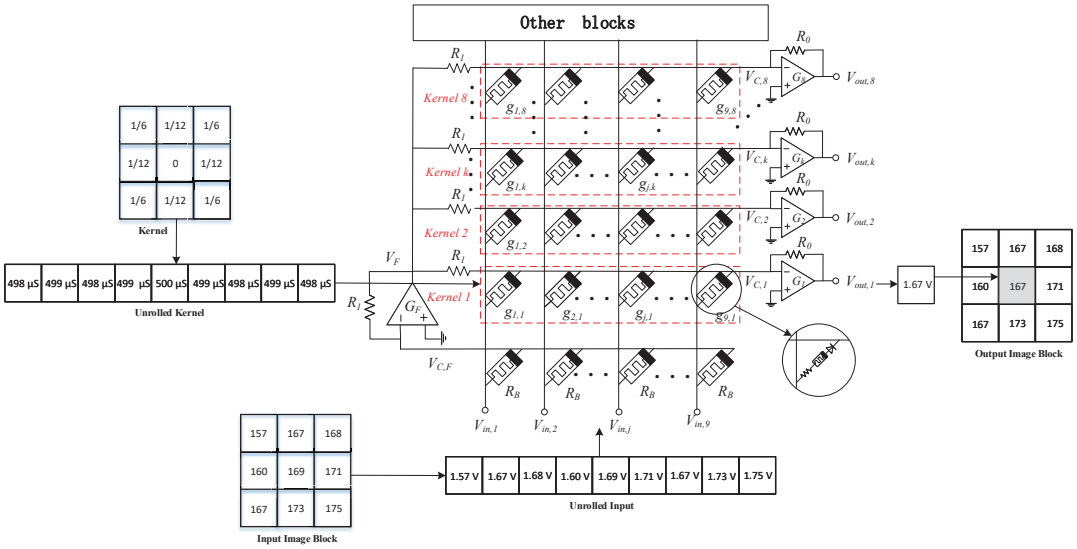


Figure 7. A memristor cross array for convolution operations; the red dotted boxes represent different convolution operators.

For the column lines, as shown in Figure 7, each row line is connected to its inverting amplifiers G_1 to G_n. For example, $V_{C,1}$ enters G_1 through negative feedback resistance R_0, $V_{out,1}$ is the output voltage of G_1, $V_{C,k}$ enters G_k similarly, and $V_{out,k}$ is the output voltage of G_k, $V_{out,k}$ can be calculated by the following Equation [33]:

$$V_{O,k} = -\left[\sum_{j=1}^{m}\left(R_0 \cdot g_{j,k} \cdot V_{IN,j}\right) + \frac{R_0}{R_1}V_F\right] \tag{20}$$

Incorporating Equation (19) into Equation (20), we can get:

$$\begin{aligned} V_{O,k} &= -\sum_{j=1}^{m}\left[R_0 \cdot g_{j,k} \cdot V_{IN,j} - \frac{R_0}{R_B}V_{IN,j}\right] \\ &= -\sum_{j=1}^{m} R_0 \cdot \left(\frac{1}{R_{j,k}} - \frac{1}{R_B}\right) \cdot V_{IN,j} \\ &= \sum_{j=1}^{m} R_0 \cdot \left(g_B - g_{j,k}\right) \cdot V_{IN,j} \end{aligned} \tag{21}$$

If $\sum_{j=1}^{m} R_0 \cdot \left(g_B - g_{j,k}\right)$ is determined by the symmetric convolution kernel F in the kth row and the jth column, then we can rewrite Equation (21) as:

$$V_{O,k} = \sum_{j=1}^{m} F_{j,k} V_{IN,j} \tag{22}$$

$$\mathrm{F}_{j,k} = R_0 \cdot \left(g_B - g_{j,k}\right) \tag{23}$$

Assuming that the conductance value range of the memristor is 0–1024 μS (where 0 means a minimal number, not 0 μS), the maximum number of digits represented by the memristor is 8 bit, so every 12 μS is equivalent to the number 1, assuming g_B is 500 μS and R_0 is 83 kΩ. As shown in Figure 7, a 3 × 3 input image block is randomly selected, and a 3 × 3 SRMC operator [2] is used for convolution operation here. First, expand the 3 × 3 SRMC operator into 1 × 9 rows, convert them into conductance values, and store them in the memristor cross-array. Then, expanding the input 3 × 3 image block pixel values into 1 × 9 rows, convert the 0–255 pixels into 0–2.55 V voltages, and input

them into the memristor array through the column lines. The final V_{out}, one output voltage is 1.67 V, and the image pixel result calculated by convolution is 167, which verifies the correctness of the circuit.

3.2. Convolution Operation on Memristor Cross-Array Circuit

The state of the spin memristor is determined by the amount of charge or magnetic flux passing through it. Within the significant charge and magnetic flux range, if the total amount of charge or total magnetic flux passing through the spin memristor is zero, the memristor will eventually return to the initial value state [37].

Figure 8 shows the state change of the memristor whose initial state is x_0 caused by a symmetrical pulse current. The amplitude of the first half of the current is $-I_A$, and the width is T_W, which converts the spin memristor state to x_1 as follows:

$$x_1 = x_0 + \frac{\Gamma_v}{h \cdot z} \cdot (-I_A \cdot T_W) \tag{24}$$

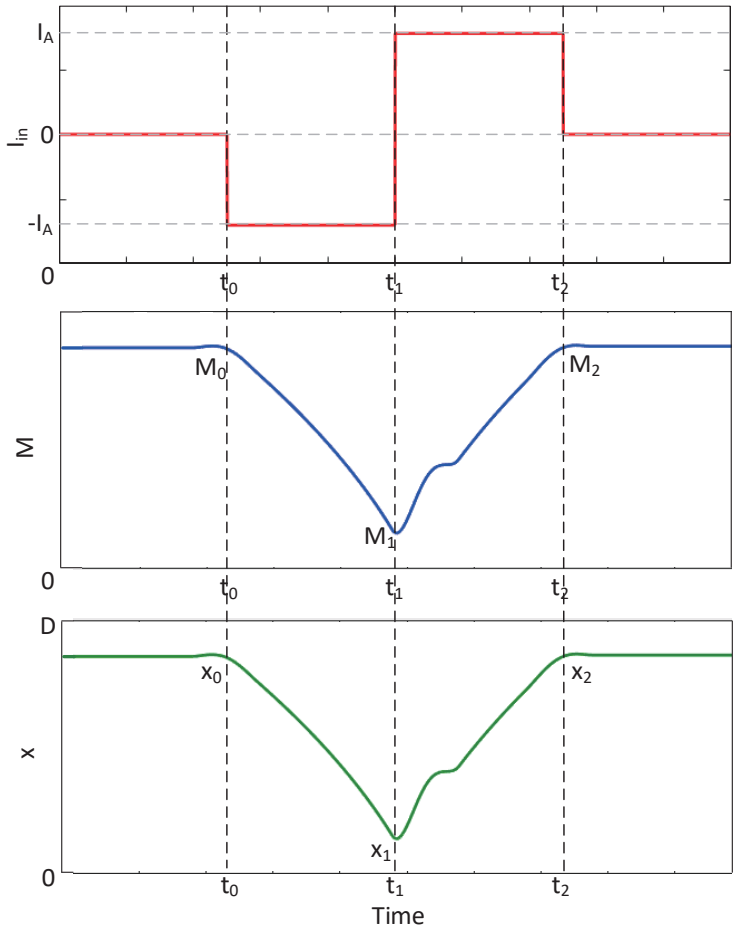


Figure 8. Symmetrical pulse current causes a change in the state of the memristor. From top to bottom are the input current I_{in}, the memristance value M, and the curve of the state variable x with time.

The amplitude of the second half current is I_A, and the width is T_W, and continue to switch the memristor state to x_2 as follows:

$$x_2 = x_1 + \frac{\Gamma_v}{h \cdot z} \cdot (I_A \cdot T_W) = x_0 \tag{25}$$

It can be seen that if the total amount of charge passing through the memristor is zero, the state of the memristor will eventually return to the initial state. If the total magnetic flux passing through the memristor is zero, the same effect will be achieved. For simplicity, suppose the power supply is a symmetrical pulse voltage with amplitudes $-V_A$ and V_A, and the widths of the two parts are both T_W. Using the relationship between the memristance value and magnetic flux in Equation (8), within the effective magnetic flux range, there are:

$$g_1 = \frac{1}{\sqrt{\frac{1}{g_0^2} - 2B\left(-V_{\mathrm{A}}T_{\mathrm{W}}\right)}} \tag{26}$$

$$g_2 = \frac{1}{\sqrt{\frac{1}{g_1^2} - 2B\left(-V_{\mathrm{A}}T_{\mathrm{W}}\right)}} = \frac{1}{\sqrt{\frac{1}{g_0^2} - 2B\left(-V_{\mathrm{A}}T_{\mathrm{W}}\right) - 2B\left(-V_{\mathrm{A}}T_{\mathrm{W}}\right)}} = g_0 \tag{27}$$

This property can be used for the read operation of the memristive memory. With this symmetrical sourse (current or voltage), the stored memristance value can be accurately read, so that the read operation will not adversely affect the stored memristance value. It is known that the resistance value of the memristor depends on the polarity, size, and time length of the external power. Assuming that the external power is a voltage pulse (amplitude is V_{W}, width is t_p), the write operation can set the memristor from the initial conductance g_0 to g' as follows:

$$g' = \frac{1}{\sqrt{\frac{1}{g_0^2} - 2B\varphi}} = \frac{1}{\sqrt{\frac{1}{g_0^2} - 2BV_{\mathrm{w}}t_{\mathrm{p}}}} \tag{28}$$

Using the symmetrical read voltage mode, the stored memristance value $g_{\mathrm{out}} = i_{\mathrm{o}}/v_{\mathrm{o}} = g'$ can be read, and then V_W is

$$V_{\mathrm{W}} = \frac{1}{2Bt_{\mathrm{p}}}\left(\frac{1}{g_0^2} - \left(\frac{v_{\mathrm{o}}}{i_{\mathrm{o}}}\right)^2\right) \tag{29}$$

As shown in Figure 9, the memristor cross array circuit that implements the convolution operation includes a spin memristor cross array, row/column address selector, and address encoder, read/write control, read circuit, weighted average filter.

In the writing mode, the color image is decomposed into three channels of R, G, and B, decomposed into multiple $N \times N$ image blocks, expanded into $1 \times N^2$ rows, and then converted into voltage input from below, range for [0 V, 2.55 V], store the convolution kernel operator into the memristor array by controlling the address selector. The salt and pepper noise in the image is removed by a weighted average filter. When constructing the filter, the switching strategy is adopted to improve the filtering effect. In fact, the mean filter is called an algorithm, which makes an image blurry. In order to solve this problem, this paper proposes a switching strategy to find pixels that may be contaminated, and the mean filter is only applicable to noisy pixels. Specifically, the algorithm using the switching strategy has two-pixel thresholds p_1 and p_2, which represent the upper threshold and the lower threshold, respectively. Set [p_1, p_2] to [5, 250], that is, [V_{tu}, V_{tl}] to [0.05 V, 2.50 V]. First, execute the read mode to determine whether the current pixel is a noise point. If it is determined to be a normal pixel, inputs a voltage signal. Suppose the pixel value exceeds the range of [p_1, p_2] (that is, the output voltage $V_{O(t)}$ exceeds the voltage range [V_{tu}, V_{tl}] in the read mode, where V_{tu} and V_{tl} are the upper and lower threshold voltages), the pixel is considered as a noise point that should be processed by the filter. In the image edge extraction application, the weighted average filter in the dashed box in Figure 9 is not needed, so this part is deleted, reflecting the flexibility of the circuit.

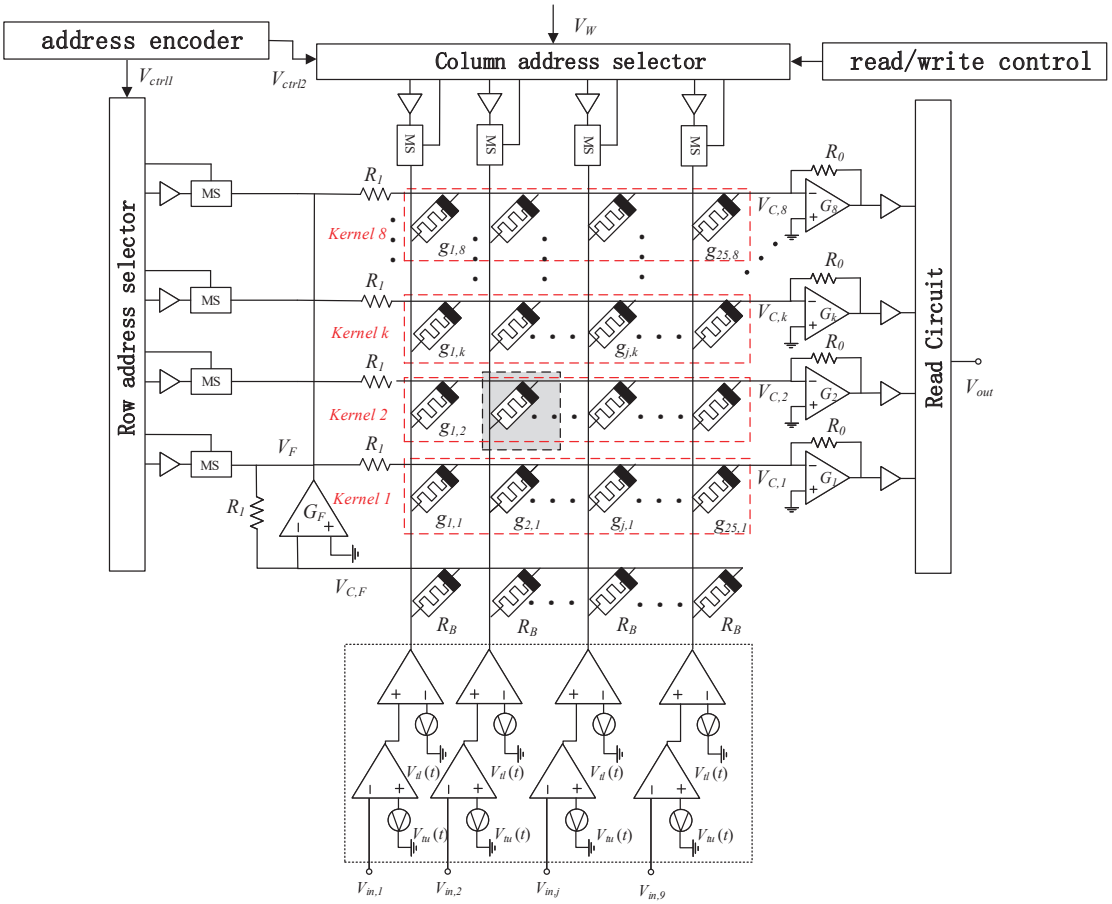


Figure 9. A memristor cross-array circuit for convolution operation. The dashed box is the weighted average filter.

Under the control of the address decoder and the row/column address selector, the input writes voltage pulse V_W is applied to the selected memristor to change its resistance state. When the write operation is over, the voltage across the memristor is zero, and the memristance value remains unchanged to realize the memory function.

In the read mode, when the read signal is valid, the read-write control circuit controls the selector to generate the symmetrical mode read voltage signal as described above and applies it to the target memristor. At this time, the read circuit can measure the flow through the memory resistor current and output it as an output signal. So far, the write and read operations of the memristor are completed.

To verify the effectiveness of the memristor simulation storage scheme proposed in this paper, computer simulations are carried out, and the memristor used is its mathematical model. The memristor parameters are set to r_L = 5 GΩ, r_H = 6 GΩ, D = 1000 nm, h = 70 nm, z = 10 nm, and J_{cr} is approximately equal to $5.74 \cdot e^{12}$. A write voltage pulse is applied to a certain memristor in the cross array, as shown in Figure 10a, and the corresponding memristance value change is shown in Figure 10c. The reading voltage as shown in Figure 10b is applied to the memristor, and the corresponding memristance value changes are shown in Figure 10d.

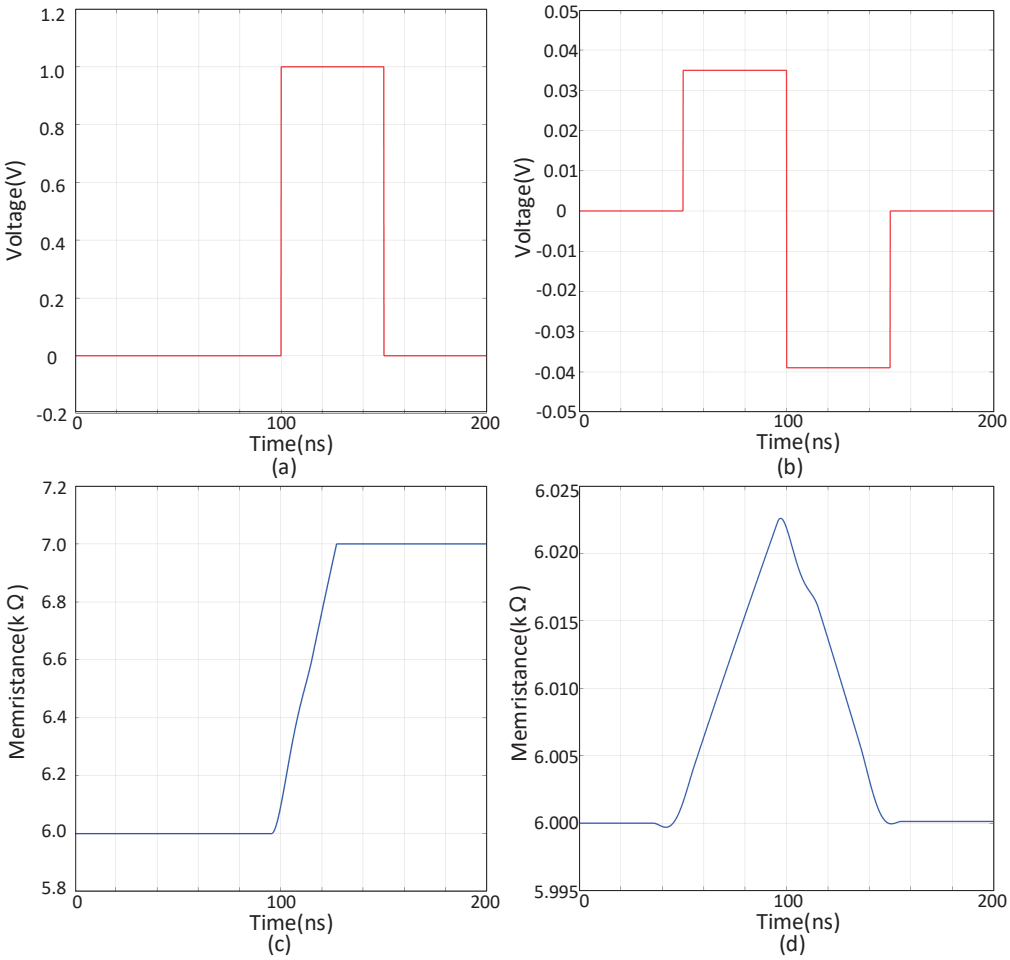


Figure 10. Write, read and restore operations of a charge-controlled spin memristor. (**a**) Write voltage pulse, (**b**) Read voltage pulse, (**c**) Write memristance value, (**d**) Read memristance value.

4. Application of Convolution Circuit in Color Image Denoising and Color Image Edge Extraction

By controlling the row/column address selector in Figure 9, 1–8 convolution operators can be stored in the memristor cross array in the form of conductance, and then the pixels of the image to be processed are converted into voltages. Voltages input from the bottom to the memristor cross array to realize multiplication and addition operations, which can realize different convolution operations, which reflects the flexibility of the circuit. This section introduces two image processing examples. In the first example, the circuit is used to achieve color image denoising and sharpening, and the filtering effects of different operators are compared. The second example is an edge detector, which uses a circuit structure that is faster and simpler than the previous circuit structure. This article uses MATLAB2020(a) version software for application-level simulation.

4.1. Color Image Denoising Based on Different Filter Operators

Figure 11 shows the convolution operator used in the weighted average filter, where the operator (a) [34] is implemented by two horizontal and vertical convolutions to store the convolution kernel operator in the memristor array in the training mode. Please note that the sum voltage is 1/12 times the read voltage, so the calculation is weighted, and no additional division is required. By continuously using the operator convolution twice, the memristor at the core of the mask can be modified to the expected state. First, assume that a memristor is a target. Set $[p_1, p_2]$ to [5, 250], that is, $[V_{tu}, V_{tl}]$ to [0.05 V, 2.50 V]. First, execute the read mode to determine whether the current pixel is a noise point. If it is determined to be a normal pixel, the value stored in the corresponding memristor

will not change. Secondly, as described in the second section, sequentially store, train, and use convolution. Finally, by moving the image block, sliding processing is realized.

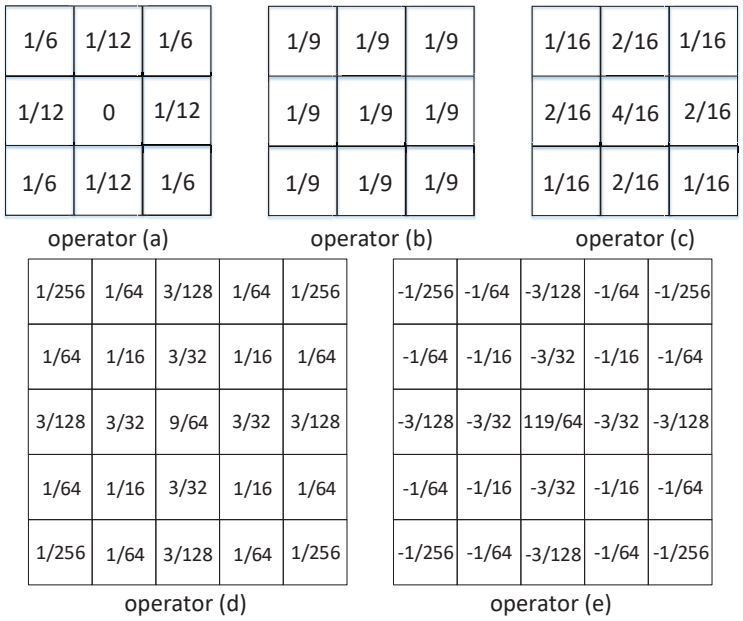


Figure 11. The convolution operator used in the weighted average filter. (**a**) SRMC operator [2]. (**b**) median filter operator. (**c**) 3 × 3 gaussian filter operator. (**d**) 5 × 5 gaussian filter operator. (**e**) image sharpening operator.

Operator (b) [38] is a median filter operator, taking the average of the nine values in the operator instead of the intermediate pixel value, so it has a smoothing and denoising effect. The gaussian filter operator presents a gaussian distribution in the horizontal and vertical directions, which highlights the weight of the center point after pixel smoothing has a better smoothing effect compared with mean filtering. Operator (c) is a 3 × 3 gaussian filter operator, operator (d) is a 5 × 5 gaussian filter operator [39], and operator (e) is an improved image sharpening operator based on the 5 × 5 gaussian filter operator. What is used is that the edge information in the image has higher contrast than the surrounding pixels, and this contrast is further enhanced after convolution, so that the image appears sharp and clear, which has the effect of sharpening the image.

Repeat these steps by applying the above operators until all pixels have removed the salt and pepper noise. We chose Lena as the primary material for the simulation. As shown in Figure 12a, 5% salt and pepper noise is added to the image in Figure 12b. Read the image that is denoised and stored in the original array in Figure 12b. Table 1 shows the power signal-to-noise ratio (PSNR) and structural similarity (SSIM) under different noise ratios. It means that the implementation has sufficient performance for different noise rates and gives correct results. Figure 12g is an improved gaussian filter operator, which achieves sharpened images. In short, the simulation results reported in this paper show that the circuit can implement a weighted average filter and give correct results.

Table 1. PSNR and SSIM for different operator.

Test Items	Operator(a)	Operator(b)	Operator(c)	Operator(d)	Operator(e)
PSNR(dB)	14.8294	15.8517	17.2997	16.8694	8.3064
SSIM	0.2569	0.4135	0.5943	0.5793	0.0481

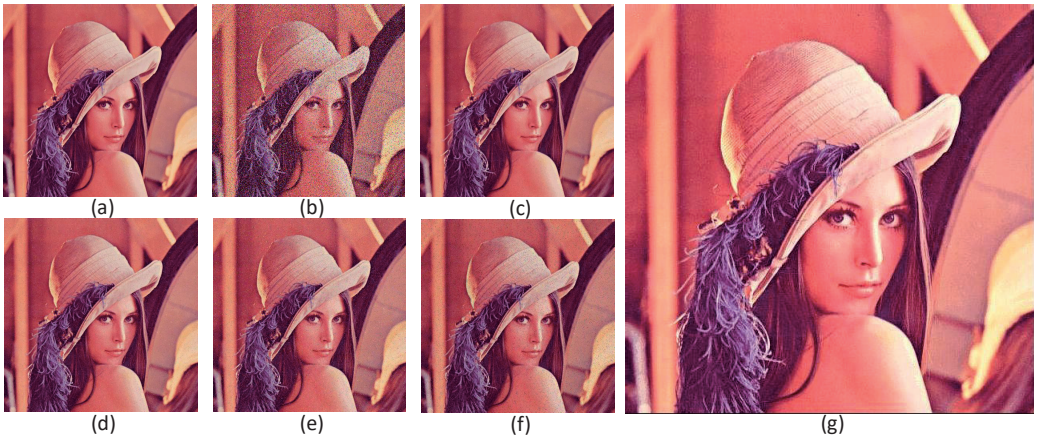


Figure 12. Filtered result. (**a**) Original image. (**b**) Lena added 5% salt and pepper noise. (**c**) Image denoised by SRMC operator. (**d**) Image denoised by median filter operator. (**e**) The image denoised by the 3 × 3 Gaussian filtering operator. (**f**) The image denoised by the 5 × 5 Gaussian filtering operator. (**g**) The image after the image sharpening operator.

4.2. Color Image Edge Detection Based on Different Convolution Operators

In the circuit of Figure 9, the weighted average filter is removed, and the input voltage directly enters the memristor array.

Figure 13 shows the operator used for color image edge extraction. Figure 13(a_1–a_4) is the prewitt operator [40], and Figure 9 (b_1–b_4) is the sobel operator [41], both of which are first-order differential operator. The former is an average filter, the latter is a weighted average filter, and the detected image edge may be more significant than 2 pixels. The advantage of these two methods is that the grayscale gradient and low-noise images have better detection effect, but the disadvantage is that the processing effect is not ideal for images with multiple complex noises. Based on this, the above two operators are improved, as shown in operators (a_1–a_8) and operators (b_1–b_8), diagonal operators in 45° and 135° directions are added on the basis of the original horizontal and vertical operators. The results are shown in Figure 14b–e, and the extracted edge details are more abundant than the original. The improved sobel operator works best and even retains some of the color details of the image.

Figure 13(c_1–c_8) is the kirsch operator [42]. The extraction result is shown in Figure 14f. Eight templates are used to convolve each pixel on the image to obtain the derivative. These eight templates represent eight directions, the maximum response to eight specific edge directions on the image, and the maximum value in the calculation (the weighted sum of 3 × 3 pixels is the sum after the corresponding position is multiplied), which is output as the edge of the image.

Figure 14 (d_1–d_8) is robert operator [43], also known as a cross differential operator. The advantage is that the positioning is accurate, and the operator is simple. The disadvantage is that to get a good effect, multiple operators are required to superpose, the calculation speed is slow, and it is more sensitive to noise. The extraction result is shown in Figure 14g, and the extraction effect is average.

Figure 14 (e_1–e_4) are laplacian operators. Laplacian operators are more sensitive to noise, so the image is generally smoothed first. Because smoothing is also performed using templates, the usual segmentation algorithms are based on laplacian operators and the smoothing operator, which combined to generate a new template. It can be seen from the extraction result of Figure 14g that in addition to the edge extraction effect, the template also has the effect of sharpening the picture.

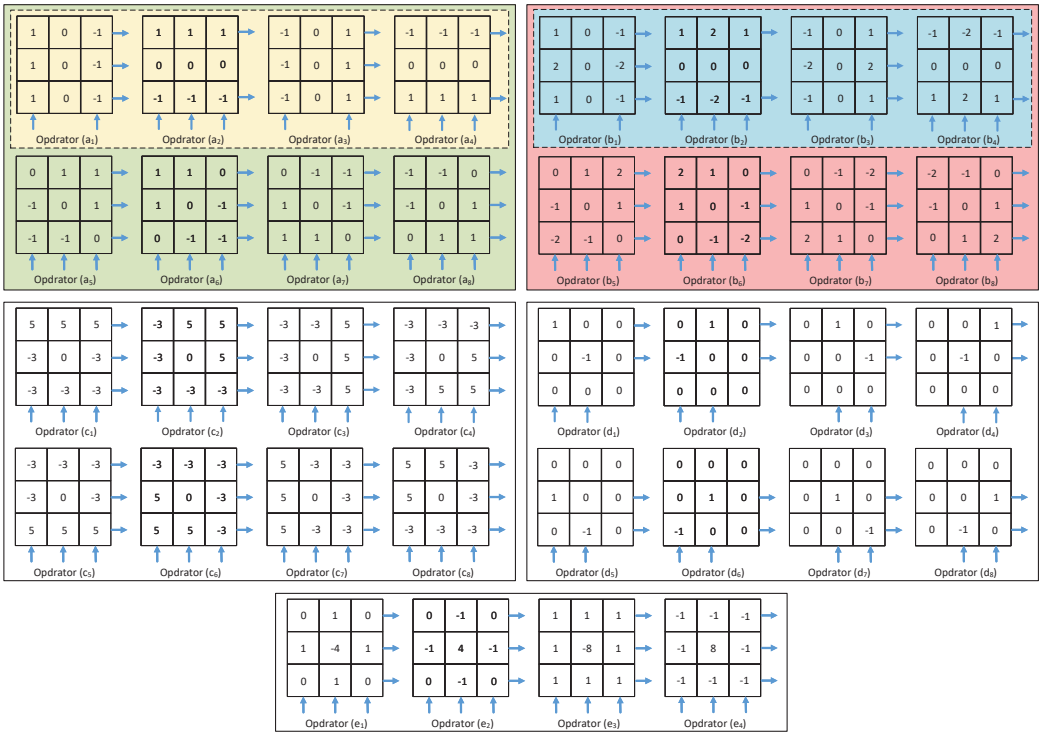


Figure 13. Convolution operator for edge detection. The blue arrow indicates the direction of current, the number sign determines the polarity of the voltage. ($\mathbf{a}_1$–$\mathbf{a}_4$) Prewitt operator (in the yellow box). ($\mathbf{a}_1$–$\mathbf{a}_8$) Proposed Prewitt operator (in the green box). ($\mathbf{b}_1$–$\mathbf{b}_4$) Sobel operator (in the blue box). ($\mathbf{b}_1$–$\mathbf{b}_8$) Proposed Sobel operator (in the red box). ($\mathbf{c}_1$–$\mathbf{c}_8$) Kirsch operator. ($\mathbf{d}_1$–$\mathbf{d}_8$) Robert operator. ($\mathbf{e}_1$–$\mathbf{e}_4$) Laplacian operator.

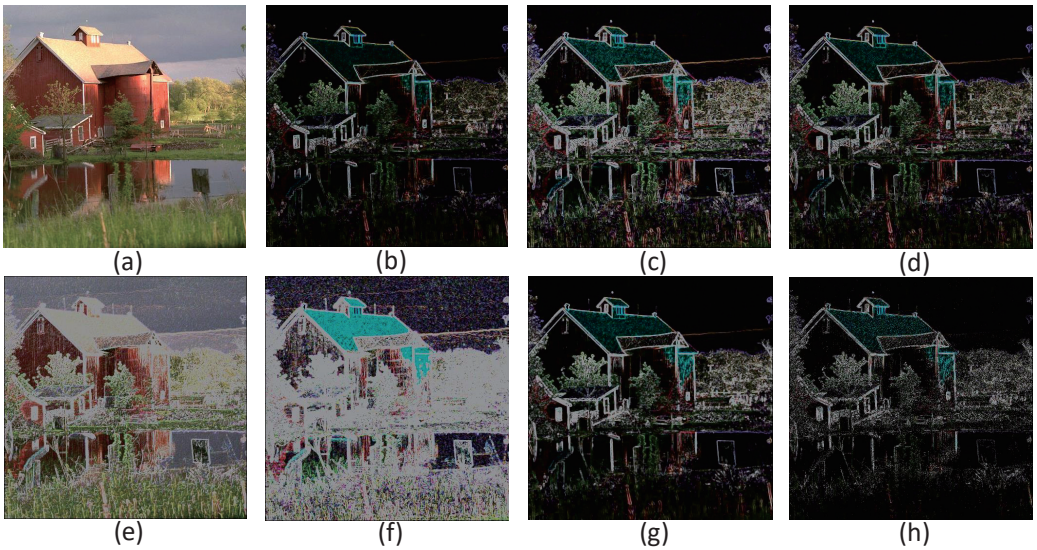


Figure 14. The realization of color image edge extraction.(**a**) Original image, (**b**) Prewitt operator, (**c**) Proposed Prewitt operator, (**d**) Sobel operator, (**e**) Proposed Sobel operator, (**f**) Kirsch operator, (**g**) Robert operator, (**h**) Laplacian operator.

Use the above operators to process the pixel values stored in the memristor array. In the case where the input circuit provides only one summation voltage at the same time, a negative number is included in the convolution. In order to avoid storing negative values (i.e., negative pixels) in the target memristor, we first use positive convolution to limit the negative result to 0 and maintain the

positive result normally. In addition, the resulting pixel value has a risk of 255. This problem can be solved by limiting the maximum (current) output of VCCS at 13.209 μA By sliding the image block repeatedly, the edge of the color image can be extracted effectively until all pixels are processed by edge detector. We chose a 256 × 256 pixel picture as the primary material for this simulation. The result is shown in Figure 14. Table 2 shows the power signal-to-noise ratio (PSNR) of the edge extraction results after processing by different operators.

Table 2. PSNR for different operator.

Test Items	Prewitt	Proposed Prewitt	Soble	Proposed Soble	Kirsch	Robert	Laplacian
PSNR(dB)	7.7960	8.8183	8.2401	15.5461	11.2648	8.4652	8.0827

5. Conclusions

This paper proposes a generalized circuit scheme based on the filtering convolution operator and the edge extraction convolution operator to implement image processing applications on the spin memristor crossover to alleviate the storage bottleneck of data-intensive applications. By adopting a self-updating circuit and parallel multi-bit selective adder and convolution algorithm, we implement color image filtering and edge extraction with different operators, and reduce the dependence on data exchange to the lowest level. All the devices used in this paper are compatible with CMOS technology, so the proposed implementation scheme also shows the advantages of large-scale integrated manufacturing. The practicability and excellent performance of this work in image processing have been proved by algorithm simulation.

Author Contributions: Conceptualization, S.Z. and Z.D.; methodology, S.Z., Z.D. and L.W.; software, S.Z.; validation, S.Z., Z.D. and L.W.; formal analysis, S.Z. and L.W.; investigation, S.Z.; resources, L.W.; data curation, S.Z. and L.W.; writing–original draft preparation, S.Z.; writing–review and editing, S.Z., Z.D. and L.W.; visualization, S.Z.; supervision, L.W., Z.D. and S.D.; project administration, L.W. and S.D.; funding acquisition, L.W. and S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (Grant No. 2018YFB1306600), the National Natural Science Foundation of China (Grant Nos.62076207, 62076208), the Fundamental Science and Advanced Technology Research Foundation of Chongqing, China (Grant Nos. cstc2017jcyjBX0050).

Acknowledgments: The authors would like to thank the National Key R&D Program of China, the National Natural Science Foundation of China, the Fundamental Science and Advanced Technology Research Foundation of Chongqing, Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, School of Electronic Information Engineering, Southwest University, and all the partners (Lidan Wang, Shukai Duan, Zhekang Dong and Chunyan Ren) for their support in the conceptualization and revision of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest. The funders support in the collection, analyses or interpretation of data; in the writing of the manuscript, or in the decision to publish the result

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism

References

1. Chen, B.; Polatkan, G.; Sapiro, G.; Blei, D.; Dunson, D.; Carin, L. Deep leaming with hierarchical convolutional factor analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1887–1901. [CrossRef]
2. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295–307. [CrossRef] [PubMed]
3. Kaxiras, S. *Architecture at the End of Moore*; Springer: New York, NY, USA, 2012.

4. Esmaeilzadeh, H.; Blem, E.; Amant, R.S.; Sankaralingam, K.; Burger, D. Dark silicon and the end of multicore scaling. In Proceedings of the 2011 38th Annual international symposium on computer architecture (ISCA), San Jose, CA, USA, 4–8 June 2011; pp. 365–376.
5. Taur, Y. CMOS design near the limit of scaling. *IBM J. Res. Dev.* **2002**, *46*, 213–222. [CrossRef]
6. Sheikh, H.F.; Ahmad, I.; Fan, D. An evolutionary technique for performance-energy-temperature optimized scheduling of parallel tasks on multicore processors. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 668–681. [CrossRef]
7. Farmahini-Farahani, A.; Ahn, J.H.; Morrow, K.; Kim, N.S. DRAMA: An architecture for accelerated processing near memory. *IEEE Comput. Archit. Lett.* **2015**, *14*, 26–29. [CrossRef]
8. Azarkhish, E.; Pfister, C.; Rossi, D.; Loi, I.; Benini, L. Logic-base interconnect design for near memory computing in the smart memory cube. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 210–223. [CrossRef]
9. Xue, W.; Yang, C.; Fu, H.; Wang, X.; Xu, Y.; Gan, L.; Lu, Y.; Zhu, X. Enabling and scaling a global shallow-water atmospheric model on Tianhe-2. In Proceedings of the 2014 IEEE 28th International Parallel and Distributed Processing Symposium, Phoenix, AZ, USA, 19–23 May 2014; pp. 745–754.
10. Zhang, X.; Yang, C.; Liu, F.; Liu, Y.; Lu, Y. Optimizing and scaling HPCG on Tianhe-2: Early experience. In Proceedings of the 14th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Dalian, China, 24–27 August 2014; pp. 28–41.
11. Bell, G.; Gray, J. What's next in high-performance computing? *Commun. ACM* **2002**, *45*, 91–95. [CrossRef]
12. Chua, L.O. Memristor—The missing circuit element. *IEEE Trans. Circuit Theory* **1971**, *5*, 507–519. [CrossRef]
13. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S. The missing memristor found. *Nature* **2008**, *453*, 80–83. [CrossRef]
14. Kvatinsky, S.; Ramadan, M.; Friedman, E.G.; Kolodny, A.; Williams, R.S. VTEAM: A general model for voltage-controlled memristors. *IEEE Trans. Circuits Syst. II Express Briefs* **2015**, *62*, 786–790. [CrossRef]
15. Chen, Y.C.Y.; Wang, X.W.X. Compact modeling and corner analysis of spintronic memristor. In Proceedings of the 2009 IEEE/ACM International Symposium on Nanoscale Architectures, San Francisco, CA, USA, 30–31 July 2009.
16. Li, C.; Hu, M.; Li, Y.; Jiang, H.; Ge, N.; Montgomery, E.; Zhang, J.; Song, W.; Dávila, N.; Graves, C.E.; et al. Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* **2017**, *1*, 52. [CrossRef]
17. Yao, P.; Wu, H.; Gao, B.; Tang, J.; Zhang, Q.; Zhang, W.; Yang, J.J.; Qian, H. Fully hardware-implemented memristor convolutional neural network. *Nature* **2020**, *577*, 641–646. [CrossRef]
18. Cai, F.; Correll, J.M.; Lee, S.H.; Lim, Y.; Bothra, V.; Zhang, Z.; Flynn, M.P.; Lu, W.D. A fully integrated reprogrammable memristor-CMOS system for efficient multiply-accumulate operations. *Nat. Electron.* **2019**, *2*, 290–299. [CrossRef]
19. Dong, Z.; Qi, D.; He, Y.; Xu, Z.; Hu, X.; Duan, S. Easily Cascaded Memristor-CMOS Hybrid Circuit for High-Efficiency Boolean Logic Implementation. *Int. J. Bifurc. Chaos* **2018**, *27*, 1850149. [CrossRef]
20. Dong, Z.; He, Y.; Hu, X.; Qi, D.; Duan, S. Flexible memristor-based LUC and its network integration for Boolean logic implementation. *IET Nanodielectr.* **2019**, *2*, 61–69. [CrossRef]
21. Dong, Z.; Lai, C.S.; Qi, D.; Xu, Z.; Li, C.; Duan, S. A general memristor-based pulse coupled neural network with variable linking coefficient for multi-focus image fusion. *Neurocomputing* **2018**, *308*, 172–183. [CrossRef]
22. Dong, Z.; Zhang, S.; Ma, B.; Qi, D.; Luo, L.; Zhou, M. A Hybrid Multi-Frame Super-Resolution Algorithm Using Multi-Channel Memristive Pulse Coupled Neural Network and Sparse Coding. In Proceedings of the 2019 7th International Conference on Information, Communication and Networks (ICICN), Macao, China, 24–26 April 2019.
23. Duan, S.; Hu, X.; Wang, L.; Li, C.; Mazumder, P. Memristor-based RRAM with applications. *Sci. China Inf. Sci.* **2012**, *55*, 1446–1460. [CrossRef]
24. Wang, L.; Li, H.; Duan, S.; Huang, T.; Wang, H. Pavlov associative memory in a memristive neural network and its circuit implementation. *Neurocomputing* **2016**, *171*, 23–29. [CrossRef]
25. Duan, S.; Wang, H.; Wang, L.; Huang, T.; Li, C. Impulsive Effects and Stability Analysis on Memristive Neural Networks With Variable Delays. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 476–481. [CrossRef]
26. Duan, S.; Hu, X.; Dong, Z.; Wang, L.; Mazumder, P. Memristor-Based Cellular Nonlinear/Neural Network: Design, Analysis, and Applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 1202–1213. [CrossRef]

27. Wang, L.; Drakakis, E.; Duan, S.; He, P.; Liao, X. Memristor Model and Its Application for Chaos Generation. *Int. J. Bifurc. Chaos* **2012**, *22*, 1250205. [CrossRef]
28. Wang, L.; Duan, S. A Chaotic Attractor in Delayed Memristive System. *Abstr. Appl. Anal.* **2012**, 2012, 726927. [CrossRef]
29. Pershin, Y.V.; Ventra, M.D. Practical approach to programmable analog circuits with memristors. *IEEE Trans. Circuits Syst. Regul. Pap.* **2010**, *57*, 1857–1864. [CrossRef]
30. Pershin, Y.V.; Ventra, M.D. Spin memristive systems: Spin memory effects in semiconductor spintronics. *Phys. Rev. B Condens. Matter* **2008**, *78*, 113309. [CrossRef]
31. Saidl, V.; Němec, P.; Wadley, P.; Hills, V.; Campion, R.P.; Novák, V.; Edmonds, K.W.; Maccherozzi, F.; Dhesi, S.S.; Gallagher, L.B.; et al. Optical determination of the Néel vector in a CuMnAs thin-film antiferromagnet. *Nat. Photonics* **2017**, *11*, 91–96. [CrossRef]
32. Zhao, W.; Ravelosona, D.; Klein, J.O.; Chappert, C. Domain Wall Shift Register-Based Reconfigurable Logic. *IEEE Trans. Magn.* **2011**, *47*, 2966–2969. [CrossRef]
33. Truong, S.N.; Min, K.S.T. New Memristor-Based Crossbar Array Architecture with 50-% Area Reduction and 48-% Power Saving for Matrix-Vector Multiplication of Analog Neuromorphic Computing. *J. Semicond. Technol. Sci.* **2014**, *14*, 356–363. [CrossRef]
34. Shang, L.; Duan, S.; Wang, L.; Huang, T. SRMC: A multibit memristor crossbar for self-renewing image mask. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2018**, *26*, 2830–2841. [CrossRef]
35. Gao, L.; Chen, P.Y.; Yu, S. Demonstration of Convolution Kernel Operation on Resistive Cross-Point Array. *IEEE Electron. Device Lett.* **2016**, *37*, 870–873. [CrossRef]
36. Zhang, Y.; Shen, Y.; Wang, X.; Cao, L. A novel design for memristor based logic switch and crossbar circuits. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2015**, *62*, 1402–1411. [CrossRef]
37. Ho, Y.; Huang, G.M.; Li, P. Dynamical Properties and Design Analysis for Nonvolatile Memristor Memories. *Circuits Syst. Regul. Pap. IEEE Trans.* **2011**, *58*, 724–736. [CrossRef]
38. Huang, T.; Yang, G.J.T.G.Y.; Tang, G. A fast two-dimensional median filtering algorithm. *IEEE Trans. Acoust. Speech Signal Process.* **1979**, *27*, 13–18. [CrossRef]
39. Haddad, R.A.; Akansu, A.N. A class of fast Gaussian binomial filters for speech and image processing. *IEEE Trans. Signal Process.* **1991**, *39*, 723–727. [CrossRef]
40. Prewitt, J.M. Object enhancement and extraction. *Pict. Process. Psychopictorics* **1970**, *10*, 15–19.
41. Farid, H.; Simoncelli, E.P. Optimally rotation-equivariant directional derivative kernels. In Proceedings of the International Conference on Computer Analysis of Images and Patterns, Kiel, Germany, 10–12 September 1997; pp. 207–214.
42. Kirsch, R.A. Computer determination of the constituent structure of biological images. *Comput. Biomed. Res.* **1971**, *4*, 315–328. [CrossRef]
43. Tippett, J.T.; Borkowitz, D.A.; Clapp, L.C.; Koester, C.J.; Vanderburgh, A., Jr. *Optical and Electro-Optical Information Processing*; Massachusetts Inst of Tech Cambridge: Cambridge, MA, USA, 1965.