SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

June 2019, Vol. 62 062408:1–062408:19
https://doi.org/10.1007/s11432-018-9863-6

Circuit design of RRAM-based neuromorphic hardware systems for classification and modified Hebbian learning

Yuning JIANG, Peng HUANG, Zheng ZHOU & Jinfeng KANG*

Institute of Microelectronics, Peking University, Beijing 100871*, China*

Received 26 September 2018/Revised 26 December 2018/Accepted 28 March 2019/Published online 7 May 2019

Abstract This paper proposes a solution to the learning of neuromorphic hardware systems based on metal-oxide resistive random access memory (RRAM) arrays, which are used as binary electronic synapses. A modified Hebbian learning method is developed to update the binary synaptic weights, and mixed-signal circuits are designed to implement the proposed learning method. The circuits are verified by SPICE, and systematic simulations are also conducted to verify the capability of the neuromorphic system to process relatively large databases. The results show that the system presents high processing speed (10^6 examples per second) for both classification and learning, and a high recognition accuracy (up to 95.6%) on the MNIST database.

Keywords RRAM, neuromorphic system, neuron circuits, Hebbian learning, pattern classification

Citation Jiang Y N, Huang P, Zhou Z, et al. Circuit design of RRAM-based neuromorphic hardware systems for classification and modified Hebbian learning. Sci China Inf Sci, 2019, 62(6): 062408, https://doi.org/10.1007/s11432-018-9863-6

1 Introduction

These years, neuromorphic systems have been attracting so much attention as a type of parallel computing systems beyond the conventional von Neumann architecture. They have exhibited remarkable competitiveness due to the massive parallelism, fault tolerance, low power consumption and capacity for adaptive learning [1–3]. In each neuromorphic system, electronic synapses are crucial, since the number of electronic synapses is usually considerable and far exceeds that of neurons. Emerging resistive random access memory (RRAM) devices are regarded as promising synaptic devices for neuromorphic systems [3]. These two-terminal devices can emulate the connection strength of synapses directly with using their own adjustable conductance. Besides, the excellent scalability to the sub-10 nanometer regime [4, 5], low energy consumption [6, 7], and good compatibility with CMOS technology [4] also make RRAM a more competitive candidate for synaptic devices in neuromorphic systems.

However, RRAM-based neuromorphic systems are still facing with some critical challenges, one of which is the hardware implementation of learning methods. For practical systems, it is necessary to update the weights of RRAM synapses through some learning processes, in which the design and implementation of learning methods can be difficult. The existing solutions can be roughly divided into two classes: backpropagation-based methods and bio-inspired methods. Backpropagation-based methods can achieve better results in terms of the overall recognition accuracy in pattern classification tasks [8, 9]. However,

* Corresponding author (email: kangjf@pku.edu.cn)

these methods generally require heavy computation and frequent direct access to memory to modify weights. Therefore, researchers often use software to calculate the synaptic weights and then store them into RRAM arrays during learning processes [8,10]. Bio-inspired methods, such as Hebbian learning [11], spike-timing-dependent plasticity (STDP) [3,12,13], and spike-rate-dependent plasticity (SRDP) [14], are more energy efficient because they are event-driven. Previous work has demonstrated Hebbian learning in RRAM-based neural logic blocks that implement simple functions and spiking neural networks (SNNs) with memristive synapses on a sub-circuit level [11, 15]. Besides, recent researches have shown that RRAM-based neuromorphic systems or SNNs are capable of handling larger tasks if proper learning algorithms are applied [16–18]. Now, more efforts are needed on RRAM-based neuromorphic systems that have achieved a high accuracy in the classification of a large dataset using bio-inspired methods. Moreover, the discussion about the considerations of the concrete circuit design for such RRAM-based neuromorphic systems is still required.

Therefore, it is needed to develop efficient learning methods and viable circuit designs that are suitable for RRAM-based neuromorphic systems. In this paper, we propose a modified Hebbian learning method for neuromorphic systems with binary RRAM synapses. The circuits of the proposed learning method are designed and validated. Then systematic simulations and analysis are conducted, the results show that with the modified Hebbian learning method and the corresponding circuits, RRAM-based neuromorphic hardware systems can achieve powerful classification and learning functionalities. By taking advantage of the inherent parallelism, the neuromorphic system is proved to be able to process 1 million examples within a second. For the task of recognizing the Modified National Institute of Standards and Technology (MNIST) database, the simulation results show that the system can achieve a high recognition accuracy (up to 95.6%), which is significantly higher than other RRAM-based neuromorphic systems with bio-inspired learning methods reported by previous publications [12]. This work makes an important step towards the practical application of RRAM-based neuromorphic hardware systems.

The rest of the paper is organized as follows. Section 2 provides an overview of our neuromorphic system. Section 3 characterizes our RRAM devices that are used as electronic synapses, then describes the proposed modified Hebbian learning method that is used to update the synaptic weights. Section 4 presents the design and simulations of the circuits in the system. Section 5 shows the systematic simulations and analysis, with the MNIST test as a verification. Finally, Section 6 is the conclusion.

2 System overview

Figure 1 shows an overview of our RRAM-based neuromorphic system that is based on the proposed learning method. The system implements a two-layer neural network that is used to classify handwritten digits. The system consists of two layers, namely, Layer 1 and Layer 2. These two layers have identical building blocks.

(1) RRAM array. Transistor-free RRAM crossbar arrays are used as electronic synapses. Each RRAM cell can act as either an excitatory synapse or an inhibitory synapse.

(2) Presynaptic neurons. Each presynaptic neuron contains a digital part (PRE-D) and an analog part (PRE-A). The digital part latches the state of the presynaptic neuron. The analog part is used as a surface between digital circuits and RRAM arrays. The analog part of a presynaptic neuron can include an E-part that connects with a column of excitatory synapses and an I-part that connects with a column of inhibitory synapses. Both the E-part and the I-part share the same digital part of a presynaptic neuron.

(3) Postsynaptic neurons. Each postsynaptic neuron contains a digital part (POST-D) and an analog part (POST-A). The analog part, which collects synaptic currents from one row of RRAM arrays in the classification mode and generates training pulses in the learning mode, is used as a surface between digital circuits and RRAM arrays.

(4) Threshold controller. A threshold controller is used to generate a global dynamic threshold for all the analog parts of postsynaptic neurons in the layer. In each layer, there is only one threshold controller.

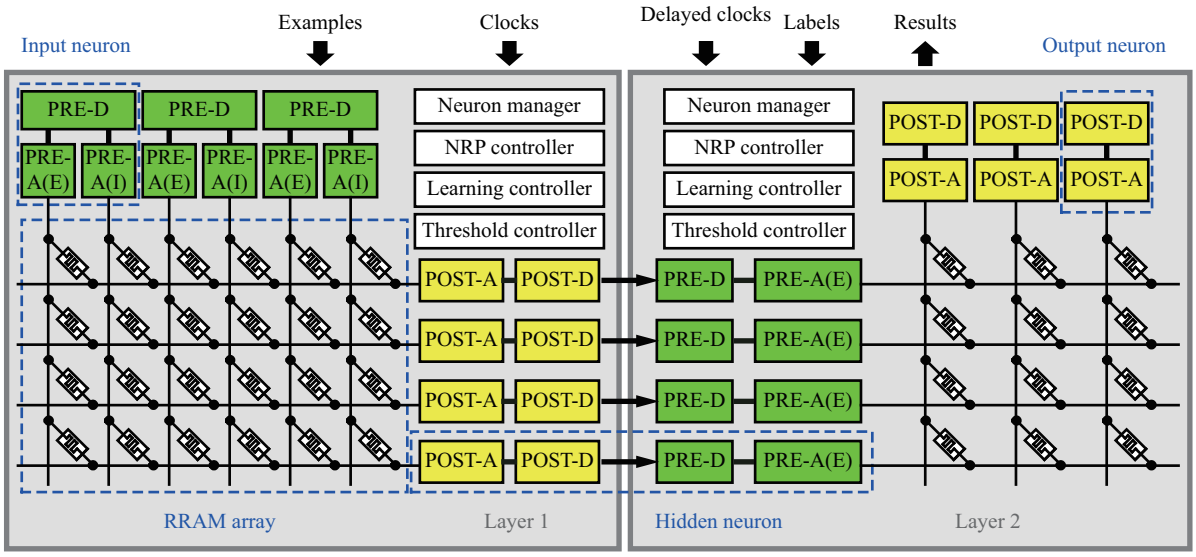

Figure 1 (Color online) An overview of the neuromorphic system in this work. The outputs of Layer 1 are directly conveyed to Layer 2 as inputs. A postsynaptic neuron in Layer 1 and the corresponding presynaptic neuron in Layer 2 form a hidden neuron together. Note that Layer 2 uses delayed clocks.

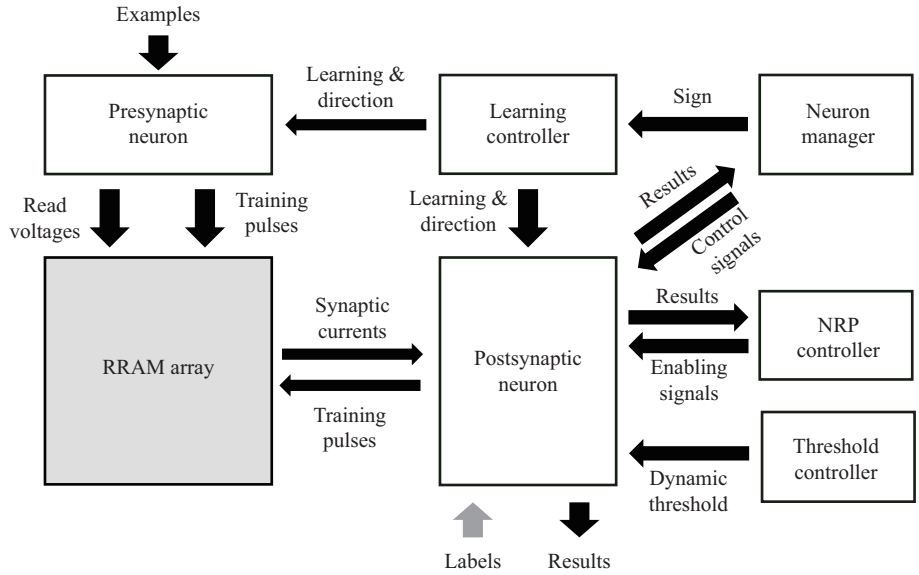

Figure 2 Communications among modules in each layer.

(5) Neuron manager. A neuron manager is a serial of digital circuits that generate control signals for digital parts of postsynaptic neurons and implement the winner-takes-all (WTA) rule, which ensures that no more than one postsynaptic neuron can fire at any time.

(6) Neuronal refractory period (NRP) controller. An NRP controller records whether a postsynaptic neuron has already fired once in the learning process, and then prevents it from firing again during the subsequent learning.

(7) Learning controller. A learning controller generates two control signals that relate to the learning process: learning and direction.

In each layer, the communications among modules are shown in Figure 2.

The output of postsynaptic neurons in Layer 1 can be used as the input of presynaptic neurons in Layer 2 directly, and no extra circuits are required. In this way, the presynaptic neurons in Layer 1 act as input neurons, the postsynaptic neurons in Layer 2 acts as output neurons, while the postsynaptic

neurons in Layer 1 and the presynaptic neurons in Layer 2 form hidden neurons together. The term "hidden" refers to the fact that these neurons are not seen directly from either the input or output of the system. For our MNIST simulations, there are 784 input neurons and 10 output neurons in the system, and we vary the number of hidden neurons in different simulations.

Several clock signals are required for the system. There are four clock signals at a 1 MHz frequency: CLK_1M, CLK_1M_D1, CLK_1M_D2, and CLK_1M_D3. CLK_1M_D1, CLK_1M_D2 and CLK_1M_D3 are created by delaying CLK_1M for 10, 20, and 30 ns, respectively. To do so, the clock signal CLK_50M at a 50 MHz frequency is also introduced. Besides, there is also a CLK_5M at a 5 MHz frequency, which is used to control the RESET/SET processes of RRAM synapses. Although both layers use a 1 MHz clock signal as the example-controlling clock signal respectively, the clock signal of Layer 2 is posterior to that of Layer 1. In our simulations, the time difference between these two example-controlling clock signals is 500 ns. In this way, these two layers work in a pipelined manner.

The entire system has been verified by SPICE simulations. The process for our SPICE simulation is the TSMC 0.18 um process. The detailed circuit design of each module is explained below.

3 Synapses and learning

3.1 RRAM synapses

In our RRAM-based neuromorphic system, transistor-free RRAM crossbar arrays are used to implement full connection between the two layers of neurons.

It is widely accepted that synapses play an important role in learning and memory. The strength of a synaptic connection is thought to result in the storage of information, which results in memory. The process of synaptic strengthening is known as long-term potentiation (LTP) [19], which implies an activity-dependent increase in the efficacy of synapses; and the opposite process is called long-term depression (LTD) [20]. To emulate these functionalities, the electronic synapses in neuromorphic systems must show plasticity. It has been proved that RRAM can show some similarities to biological synapses [3]. The conductance of RRAM can directly emulate the strength of a synapse. More importantly, the conductance is adjustable when proper external voltages are applied, and is retentive when no voltage is applied or the voltage amplitude is not large enough to cause a resistive switching event. These properties make it possible for RRAM synapses to acquire (learn) and store (memorize) information carried by the incoming flux of stimuli, just like what biological synapses do.

However, there are some problems. Bio-inspired learning methods generally require the synapses to carry real-number weights, which implies that the neuromorphic system must control the conductance of RRAM synapses precisely. As a result, previous work generally requires synaptic devices to show continuous conductance change [11, 15], or at least multilevel characteristics [12]. In fact, this is difficult because almost all the reported RRAM synapses suffer from an issue of nonlinear weight update [8]. Besides, this may also require a higher uniformity of devices. Binary RRAM synapses will be more reliable. In this work, RRAM devices work in a binary way, which loosens the requirements of realistic devices.

In our modified Hebbian learning method, there are two types of synapses: excitatory synapses and inhibitory synapses. An excitatory synapse is a synapse in which an action potential in a presynaptic neuron increases the probability of an action potential occurring in a postsynaptic neuron, while an inhibitory synapse is the opposite. Previous work on STDP simulations has proved the capability of SNNs with inhibitory units to achieve high performance on the MNIST database [21]. Figure 3 illustrates an analogy between the biological synapses and the RRAM synapses. In traditional descriptions of neural organization, a biological synapse is a connection that can impose excitation or inhibition, but not both on the postsynaptic neuron [22]. Whether a biological synapse is excitatory or inhibitory depends on the type of receptors and neurotransmitter employed at the synapse. In the RRAM implementation of synapses, to include these two types of synapses, we consider two adjacent RRAM cells as one synapse group. In each group, one RRAM cell acts as the excitatory synapse, while the other acts as the inhibitory

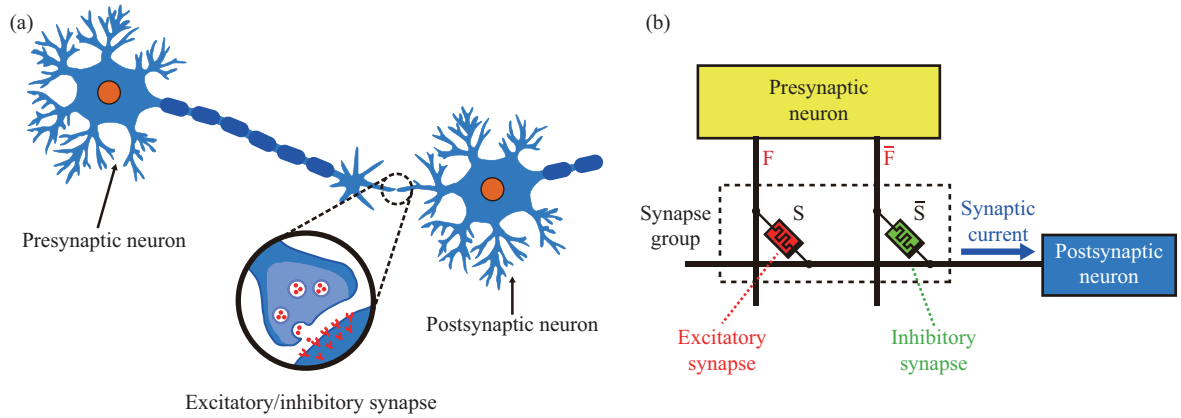

Figure 3 (Color online) An analogy between biological synapses and RRAM synapses. (a) A biological synapse; (b) RRAM synapses. In our modified Hebbian learning method, there are two types of synapses: excitatory synapses and inhibitory synapses.

Table 1 Synaptic current levels

Presynaptic neuron		Fire	Rest
Excitatory synapse	LRS	1	0
	HRS	0	0
Inhibitory synapse	LRS	0	1
	HRS	0	0

synapse. These two RRAM cells are identical in terms of polarities and structures, however, they differ in meanings and operations. For both excitatory and inhibitory synapses, the high resistance state (HRS) of RRAM represents a weak synaptic connection, while the low resistance state (LRS) of RRAM represents a strong synaptic connection. Only a strong synaptic connection can conduct synaptic current. One important difference is that excitatory synapses conduct synaptic current when the presynaptic neurons fire, whereas inhibitory synapses conduct synaptic current when the presynaptic neurons rest. They always take opposite actions. The behavior of synapses is concluded in Table 1.

Current level "1" indicates that there is synaptic current, while "0" indicates that there is not.

Note that the excitatory synapse and the inhibitory synapse in the same group can never be in LRS at the same time. In addition, if no inhibitory synapses are required, one of the RRAM cells in each synapse group can be removed. By introducing the concepts of excitatory and inhibitory synapses, we increase the diversity of synapse behaviors, and intensify the function of the neuromorphic system, as we are going to demonstrate later in this paper.

In a neuromorphic system that is based on the modified Hebbian learning, there are two operating modes: the learning mode and the classification mode. The learning mode uses training examples to induce proper changes in the synaptic weights, while the classification mode classifies testing examples and does not change any synaptic weights. The learning mode involves two types of operations of the synapses: LTD and LTP. For both excitatory and inhibitory synapses, LTD is implemented by the RESET processes of RRAM, and LTP is implemented by the SET processes of RRAM. The LTP of excitatory synapses follows the original Hebbian learning rule: "cells that fire together, wire together". In other words, LTP takes place in the excitatory synapses that connect with a firing presynaptic neuron and a firing postsynaptic neuron. For inhibitory synapses, LTP takes place in those that connect with a resting presynaptic neuron and a firing postsynaptic neuron. On the contrary, LTD is the only way to weaken the strength of a synapse. In the proposed method, LTD is controlled by either a firing presynaptic neuron or a firing postsynaptic neuron, but not both of them. If pre-controlled LTD is adopted, a firing presynaptic neuron applies LTD to each synapses it connects with before LTP. If post-controlled LTD is adopted, a firing postsynaptic neuron applies LTD to each synapses it connects with before LTP.

As for the firing of postsynaptic neurons, a WTA rule is adopted. No more than one postsynaptic

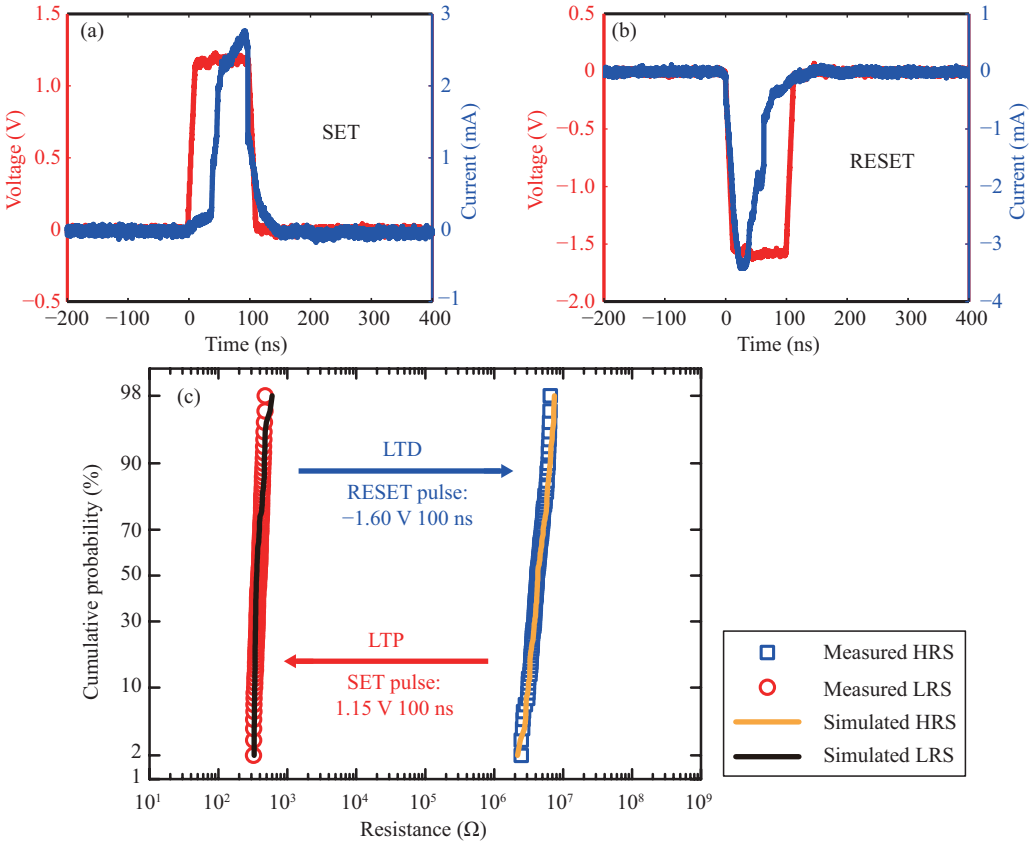

Figure 4 (Color online) (a) Measured transient response of an RRAM device during a SET process. The device switches from HRS to LRS. (b) Measured transient response of an RRAM device during a RESET process. The device switches from LRS to HRS. (c) Measured and simulated resistance distribution of the fabricated RRAM devices. The devices have two stable resistance states and show reliable LTD and LTP behaviors under successive RESET and SET pulses. The measured results indicate that the RRAM devices can be used as qualified binary synapses.

neuron can fire at any time. The firing postsynaptic neuron is identified as the "winner".

Note that our synapses are binary synapses, which implies that their synaptic weights cannot accept gradual changes. If a synapse participates in learning, it will either remain unchanged or change completely. Therefore, in the proposed learning method, new synaptic weights always replace existing synaptic weights completely. When the system is learning multiple examples, there may be an extreme case: one of the postsynaptic neurons becomes excessively active due to the relatively strong synaptic connections, and then it fires frequently, which prevents other postsynaptic neurons from learning. It may also overwrite the synapses it connects with frequently, which causes a huge loss of information. To handle this, an NRP is introduced. When a postsynaptic neuron fires, its NRP starts and it will not fire again in the following learning processes.

3.2 Array operations

In our learning method, two basic requirements should be met by the RRAM devices.

(1) RRAM should show bi-level stable resistance states to represent a strong synaptic connection and a weak synaptic connection, respectively.

(2) RRAM should be able to change its resistance state to either HRS or LRS while proper operational voltages are applied.

In this work, TiN/HfO_x/AlO_x/Pt RRAM devices are fabricated and then measured on Agilent instruments. The size of our RRAM device is 5 μm × 5 μm. The measured transient responses of an RRAM device during a SET process and a RESET process are shown in Figure 4(a) and (b), respectively. As shown in Figure 4(c), we have experimentally demonstrated that our RRAM synapses have two stable

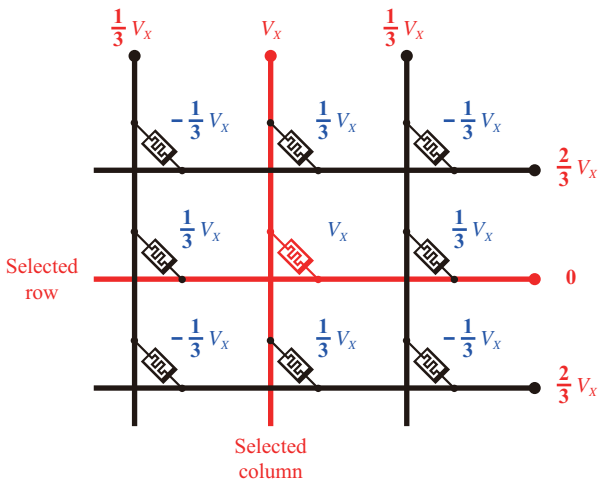

Figure 5 (Color online) Schematic diagram of the 1/3 bias scheme used in this work. The RRAM cell in the middle is selected, while others are not. By applying operation voltages (V_X and 0 on the selected columns and rows, $\frac{1}{3}V_X$ and $\frac{2}{3}V_X$ on other columns and rows) on each column and each row simultaneously, the selected RRAM cell can be switched to either HRS or LRS.

resistance states and show reliable LTD and LTP characteristics under negative and positive pulses, respectively. In the experiments, the bottom electrode of RRAM is grounded, and then positive pulses (1.15 V, 100 ns) and negative pulses (−1.60 V, 100 ns) are applied to the top electrode of RRAM alternately, and the resistance of RRAM is measured after each pulse. Two stable resistance states are presented. For LRS resistances, the coefficient of variation is 3.46%. It can be observed that the device is able to change its resistance state from HRS to LRS when a single SET pulse (1.15 V, 100 ns) is applied, and from LRS to HRS when a single RESET pulse (−1.60 V, 100 ns) is applied.

Transistor-free RRAM crossbar arrays are much easier to scale than those with an additional transistor at each cross point [4,5], thus they can achieve higher integration density. However, without any selection devices, the sneak path problem can become severe. Therefore, in the learning mode, a special bias scheme has to be adopted in the array operations to ensure a maximum success rate of learning and avoid unintended resistive switching events. In array operations, a 1/3 bias scheme is adopted [23]. As shown in Figure 5, when the RRAM cell in the middle is going to be operated, the column and the row it locates in are selected. For the selected column, voltage V_X is applied, where V_X is either the SET voltage $V_{\rm SET}$ or the RESET voltage $V_{\rm RESET}$. For the unselected columns, voltage $\frac{1}{3}V_X$ is applied. For the selected row, voltage 0 is applied, and for the unselected row, voltage $\frac{2}{3}V_X$ is applied. In this way, the voltage across the selected cell is V_X, which is enough to operate the cell, while the voltage across any unselected cell is $\pm\frac{1}{3}V_X$, which is not enough to cause any changes. Note that multiple rows and multiple columns can be selected simultaneously, therefore, parallel array operations are possible. In the classification mode, the systems take advantage of the nature of the RRAM crossbar structure to do matrix-vector multiplications in parallel [24].

3.3 Modified Hebbian learning

To use binary RRAM devices as electronic synapses in a neuromorphic system, a modified Hebbian learning method is proposed. Hebbian learning [25], which is the oldest and most famous of all learning rules, is often summarized as "cells that fire together, wire together" [26]. In many literatures [27–29], Hebbian learning is explained as a two-part rule, which contains a potentiation aspect and a depression aspect.

(1) If two neurons on either side of a synapse (connection) are activated simultaneously (i.e., synchronously), then the strength of that synapse is selectively increased.

(2) If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.

The simplest form of Hebbian learning can be formulated by [27]

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n), \tag{1}$$

where $\Delta w_{kj}(n)$ is the increment of a synaptic weight at time-step n. η is a positive constant that determines the rate of learning. The presynaptic signal and the postsynaptic signal are denoted by $x_j(n)$ and $y_k(n)$, respectively.

It can be seen that the weight updating of a synapse in Hebbian learning depends on the signal (i.e., firing or activation) of the neurons on its both sides. This helps reduce the complexity of a neuromorphic system. However, it can be shown that for any neuron model, Hebbian learning is unstable [30]. It still needs some modification to fit into RRAM-based neuromorphic systems and achieve high performance in classification tasks.

There are several important differences between the proposed modified Hebbian learning method and the conventional Hebbian learning method. In our method:

(1) All synaptic weights are binary;

(2) Two types of synapses are used: excitatory synapses and inhibitory synapses;

(3) In weight updating, LTD is applied before LTP;

(4) In weight updating, only the synapses that are connected with one neuron can be updated at a time;

(5) The method is adaptable to both supervised learning and unsupervised learning.

Details are explained below.

Figure 6 presents a flowchart that describes the main operations of the neuromorphic system based on the modified Hebbian learning.

First, the neuromorphic system checks its operating mode. When the system works in the learning mode, the NRP function is optional. If the NRP function is enabled, the system will reset the NRP of all the postsynaptic neurons first. In other words, all the postsynaptic neurons are allowed to fire at first. Then the system starts to accept an example. The example must be represented as binary data, and each bit of the data is conveyed to a presynaptic neuron. The presynaptic neuron fires when it accepts "1", and stay in rest when it accepts "0". Then the system checks whether unsupervised learning or supervised learning is adopted. In unsupervised learning, the postsynaptic neurons start a fair competition and eventually one of them fires, and then the system prevents other postsynaptic neurons from firing. In supervised learning, the label of the example is given, and the system forces the corresponding postsynaptic neuron to fire, and then prevents other postsynaptic neurons from firing. If the NRP function is enabled, the system updates the NRP of the firing postsynaptic neuron, which means the neuron is not able to fire again in the following learning processes. After that, the system begins to modify the synapses. According to the setting of the system, either pre-controlled LTD or post-controlled LTD is applied. Then LTP is applied according to the rules mentioned above. By now, the system has finished the learning of an example. Then another example is accepted and the system starts the next cycle. The cycle repeats until no more examples are applied.

An example of the transitions of RRAM cells during the modified Hebbian learning is shown in Figure 7. For the RRAM cells, the operations can be summarized as three steps: example input, LTD, and LTP. In the first step, the data of an input example is applied to the presynaptic neurons, and synaptic currents are created to make one of the postsynaptic neurons fire. Then the system begins to change the resistance states of RRAM cells. During the LTD process, all the RRAM cells that connect with the firing postsynaptic neuron are RESET to HRS. Note that RRAM cells on other rows stay unchanged. During the LTP process, the system applies SET operations to each synapse group. If the presynaptic neuron they connect with is firing, the system applies SET operations to the excitatory synapse, and the synapse group becomes an LRS-HRS pair. On the contrary, if the presynaptic neuron they connect with is not firing, the system applies SET operations to the inhibitory synapse to make an HRS-LRS pair.

Now that the system has finished learning, it can operate in the classification mode. An example is input, and then a certain postsynaptic neuron fires and becomes the winner, the system prevents other

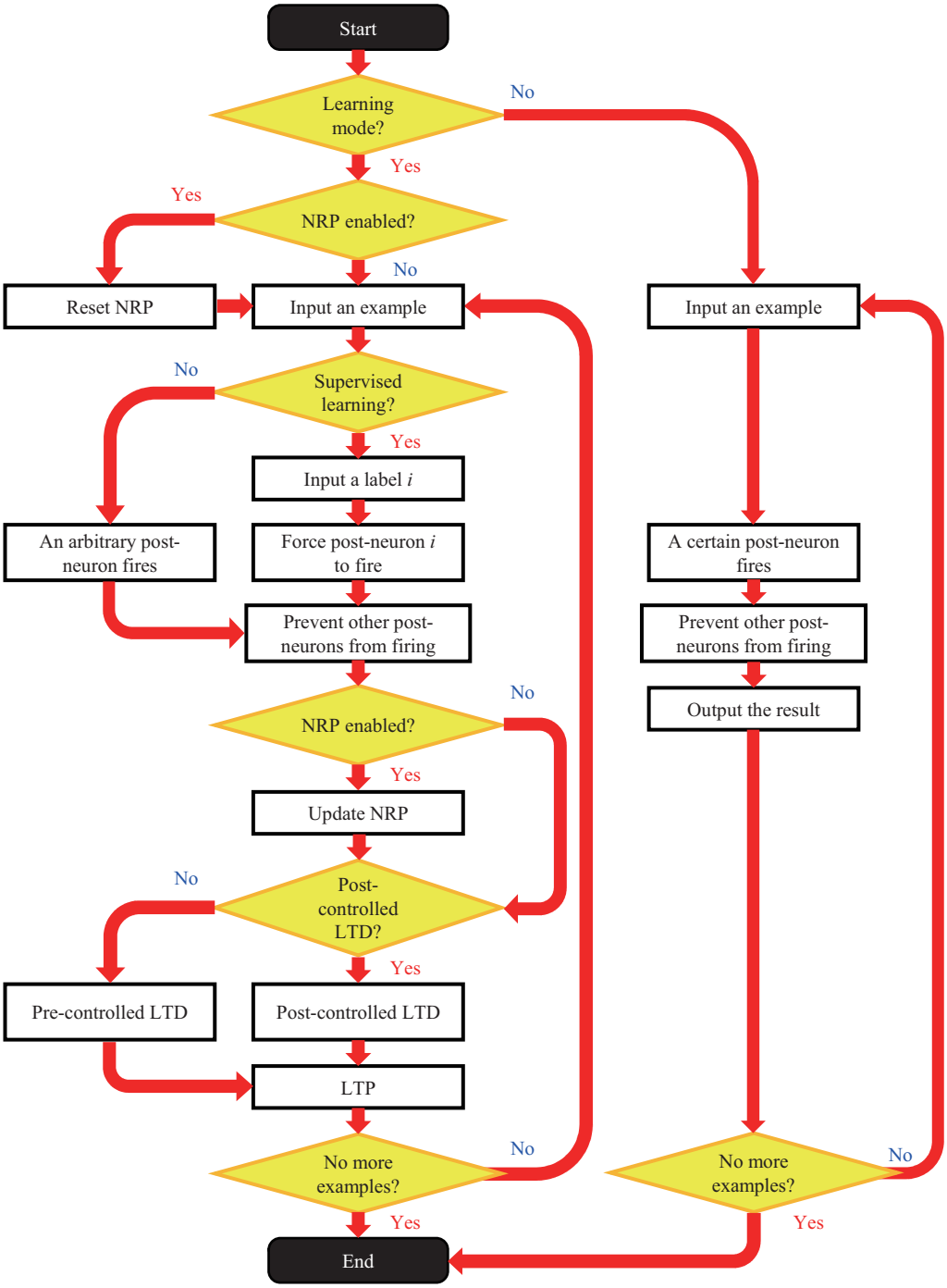

Figure 6 (Color online) Flowchart for the neuromorphic system based on the modified Hebbian learning.

postsynaptic neurons from firing. At this time, the firing postsynaptic neuron represents the classification result. Then the system outputs the result and processes the next example.

4 Circuit design

4.1 Analog parts of neuron circuits

In this work, CMOS-based analog circuits are designed to implement the analog parts of neurons, as shown in Figure 8. The function of analog parts of neurons is to connect with RRAM arrays and act as

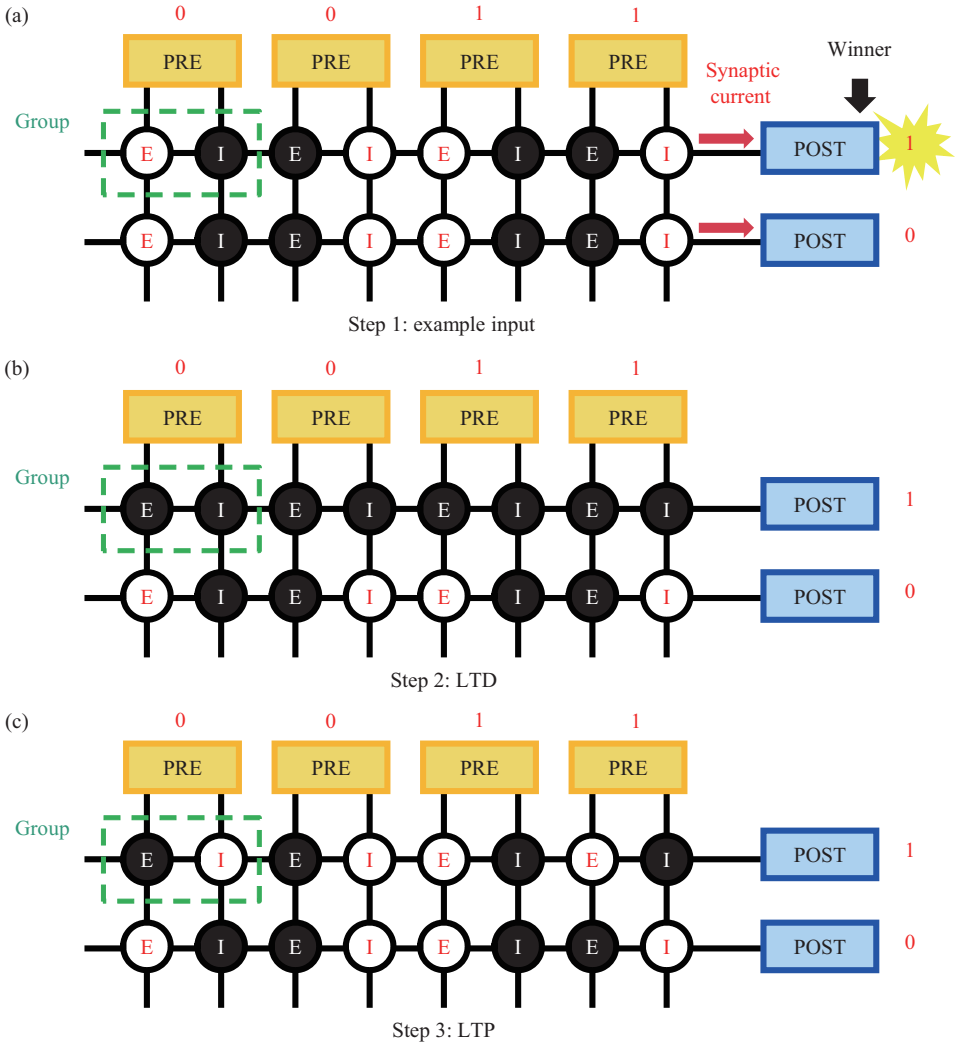

Figure 7 (Color online) An example of the transitions of RRAM cells during the modified Hebbian learning. (a) Example input; (b) LTD; (c) LTP.

an interface between the RRAM arrays and digital circuits. For the analog parts of presynaptic neurons, they apply read voltages to the arrays in the classification mode and apply training pulses to the arrays in the learning mode. For the analog parts of postsynaptic neurons, they accept synaptic currents from the arrays in the classification mode, and apply training pulses to the arrays in the learning mode. Since the RRAM arrays must be operated by certain voltages in the learning mode, they have to work with analog signals. Besides, the results of matrix-vector multiplications are represented by current intensity, they should be accepted by some analog circuits first. Therefore, analog parts of neuron circuits are essential.

The analog part of a presynaptic neuron is illustrated in Figure 8(a). STI is the stimulation signal generated by the digital part, Learn is the control signal that enables array operations, Direction is the control signal that determines whether LTD or LTP is to be applied, and CON indicates whether pre-controlled LTD or post-controlled LTD is used. Besides, there are voltage sources that are required for array operations. V_{SET3} is the proper voltage to perform SET operations (representing LTP), while V_{RESET3} is the proper voltage to perform RESET operations (representing LTD). V_{SET1} is 1/3 of V_{SET3}, and V_{RESET1} is 1/3 of V_{RESET3}. V_{Read} is a read voltage that is equal to the output of a firing presynaptic neuron, which will not cause any changes in the states of RRAM synapses. Finally, V_{PRE} is the output of the analog part of the presynaptic neuron. For E-parts and I-parts, their circuits are identical, the only difference is one of the input signals. On the top left corner of Figure 8(a), an E-part accepts the

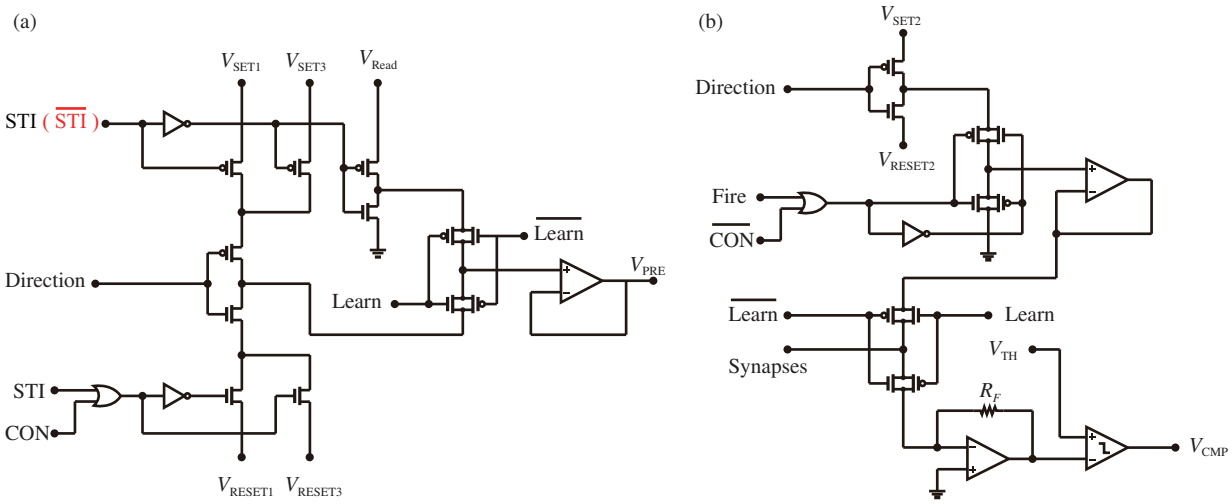

Figure 8 (Color online) Analog parts of the proposed neuron circuits. (a) The analog part (either an E-part or an I-part) of a presynaptic neuron. $V_{\rm PRE}$ is the output of the analog part of the presynaptic neuron and connects to a column of the RRAM array. (b) The analog part of a postsynaptic neuron. Synapses is an I/O port that links to a row of the RRAM array.

stimulation signal STI, while the I-part accepts the complement of STI. With all the input control signals and voltage sources, the circuit generates a suitable action potential for the presynaptic neuron, and the voltage follower ensures the neuron to drive a larger number of synapses.

The design of an analog part of a postsynaptic neuron is shown in Figure 8(b). In this circuit, Fire is a signal from the digital part of the postsynaptic neuron, it indicates whether the postsynaptic neuron fires or not. Synapses is an I/O port that links to a row of the RRAM array, $V_{\rm SET2}$ is 2/3 of $V_{\rm SET3}$, and $V_{\rm RESET2}$ is 2/3 of $V_{\rm RESET3}$. $V_{\rm TH}$ is the global dynamic threshold voltage, and $V_{\rm CMP}$ is the output of the comparator. The inverting amplifier accepts the total synaptic current and generates a local voltage that is in direct proportion to it. Then the comparator compares the local voltage with a gradually rising threshold. As soon as the threshold exceeds the local voltage, the comparator outputs a high level, which means the postsynaptic neuron makes a request of firing. The larger the synaptic current is, the earlier the postsynaptic neuron makes the request. The first postsynaptic neuron that makes the request will fire, according to our WTA rule. The analog parts of postsynaptic neuron circuits do not have to reflect precise voltage or current value to the subsequent digital circuits, thus the use of analog-to-digital converters (ADCs) is avoided. Besides, this design reduces the use of capacitors in each neuron by taking the advantage of a shared threshold controller, which will be described later.

The transient simulations of analog parts of the proposed neuron circuits are shown in Figure 9. Three examples are applied in turn. Consider one of the postsynaptic neurons and two synapses it connects with: RRAM_1 and RRAM_2. RRAM_1 is an excitatory synapse and RRAM_2 is an inhibitory synapse. After applying the first example, the postsynaptic neuron fires, and then it enters its NRP, ignoring the following two examples. Therefore, the RRAM synapses that connect with this postsynaptic neuron are operated in the first cycle. The pulses from a presynaptic neuron and the pulses from the postsynaptic neuron act on an RRAM synapse simultaneously, and the overlapped pulses can change the state of the RRAM synapse. RRAM_1 is RESET at around $t = 500$ ns, and then stays in HRS; RRAM_2 is RESET and then is immediately SET, and then stays in LRS. The waveform indicates that the states of RRAM synapses are updated under the pulses from neuron circuits.

4.2 Threshold controller

As mentioned above, a gradual rising threshold is needed in the analog parts of postsynaptic neurons. The threshold triggers the postsynaptic neurons in the order of their local voltage amplitudes. We have designed a simple threshold controller to generate this threshold, as shown in Figure 10(a). The circuit

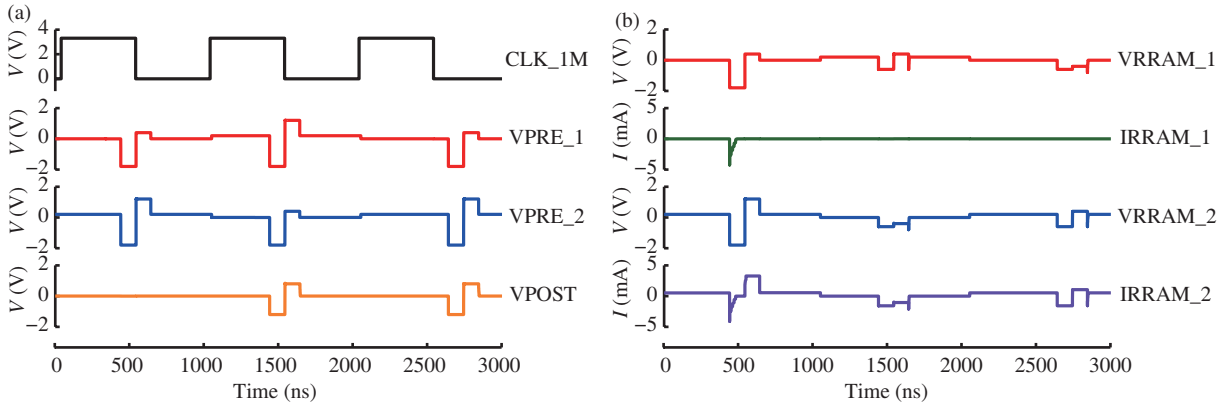

Figure 9 (Color online) The SPICE transient simulations of analog parts of the proposed neuron circuits. The overlapped pulses from both sides can change the state of the RRAM synapse. At about $t = 500$ ns, RRAM_1 changes to HRS, while RRAM_2 changes to HRS first, and then back to LRS. Then the two cells keep their resistance states unchanged during the subsequent operations. (a) Waveforms of signal CLK_1M, VPRE_1, VPRE_2 and VPOST; (b) waveforms of signal VRRAM_1, IRRAM_1, VRRAM_2 and IRRAM_2.

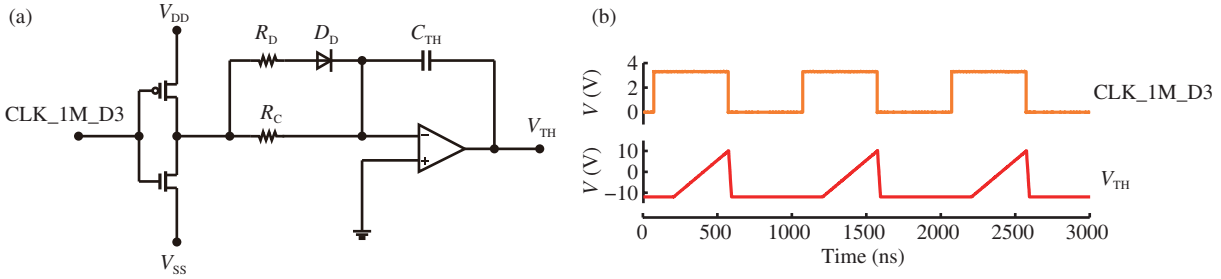

Figure 10 (Color online) (a) Circuit design of a threshold controller; (b) the SPICE transient simulations of the threshold controller. A clock signal is applied to the threshold controller, and a dynamic threshold that rises gradually and declines rapidly is generated.

consists of two MOSFETs, two resistors, one diode, and one integrator. The threshold is controlled by the clock signal CLK_1M_D3.

The transient simulations of the threshold controller are shown in Figure 10(b). When CLK_1M_D3 is high, V_{TH} rises gradually. When CLK_1M_D3 changes from high to low, V_{TH} declines rapidly.

4.3 Digital parts of neuron circuits and the neuron manager

Digital parts of neuron circuits determine whether a presynaptic or postsynaptic neuron fires or not. Besides, a neuron manager module is also required to generate control signals for digital parts of postsynaptic neurons and implement the WTA rule, which ensures that no more than one postsynaptic neuron can fire at any time.

In the digital circuits, clock signal CLK_1M_D1, CLK_1M_D2 and CLK_1M_D3 are created by delaying CLK_1M for 10, 20, and 30 ns, respectively. CLK_1M controls the NRP controller, which should be renewed before applying each example. CLK_1M_D1 is called the example-controlling clock, since the data of an example is accepted by the system at each positive edge of CLK_1M_D1. CLK_1M_D2 ensures all neurons are in the resting state before the dynamic threshold rises. CLK_1M_D3 is used to control the dynamic threshold and the input of labels, which come last.

The digital part of a presynaptic neuron is implemented by only a positive-edge-triggered D flip-flop, as shown in Figure 11(a). CLK_1M_D1 controls the transmission of the data of an example. The output Q makes the stimulation signal STI, which is conveyed to the E-part of the presynaptic neuron; while the output $\overline{Q}$ makes the complement of signal STI, which is conveyed to the I-part of the presynaptic neuron.

The digital parts of postsynaptic neurons are shown in Figure 11(b). These digital circuits accept the

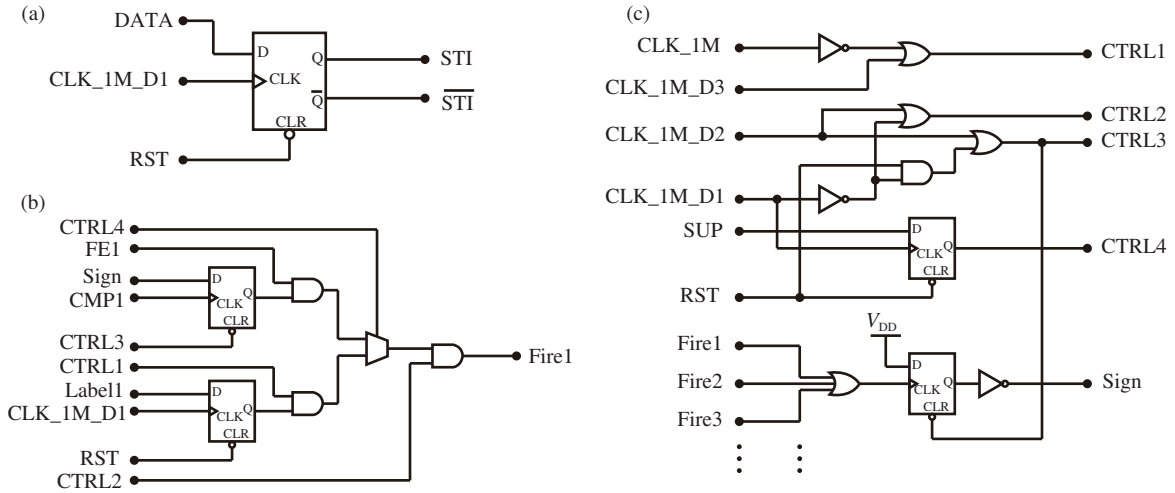

Figure 11 Digital parts of the proposed neuron circuits and the neuron manager module. (a) The digital part of a presynaptic neuron. The digital part of a presynaptic neuron is implemented by only a positive-edge-triggered D flip-flop. (b) The digital parts of a postsynaptic neuron. These digital circuits accept the output signal from the comparator in the analog part of the neuron, and accept a serial of control signals, and finally output the firing state of the neuron. (c) Neuron manager. The neuron manager generates control signals to the digital parts of postsynaptic neurons, and implements the WTA rule by sending a Sign signal that implies whether a winner neuron occurs.

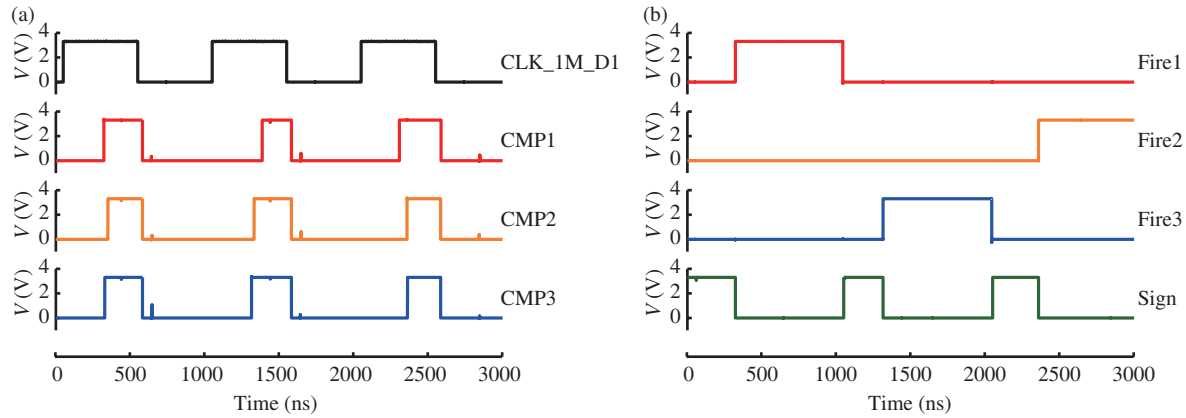

Figure 12 (Color online) The SPICE transient simulations of digital parts of the proposed neuron circuits during unsupervised learning. No label is needed in unsupervised learning, and the postsynaptic neurons compete with each other until one of them becomes the winner and fires. (a) Waveforms of signal CLK_1M_D1, CMP1, CMP2, and CMP3; (b) waveforms of signal Fire1, Fire2, Fire3, and Sign.

output signal from the comparator in the analog part of the neuron, and accept a serial of control signals, and finally output the firing state of the neuron. CMP1 is the voltage level of the comparator output in the analog part, FE1 is the firing enable signal that comes from the NRP controller module, and Label1 is the label of the input example. Sign is the signal that indicates whether the winner neuron of the cycle has been determined. CTRL1 to CTRL4 are the control signals generated by the neuron manager. Digital parts of postsynaptic neurons are based on asynchronous circuits, which cooperate well with both the neuron manager module and the analog parts of postsynaptic neurons.

The design of the neuron manager module is shown in Figure 11(c). SUP indicates whether the system performs supervised learning or not. The global clock with 1 MHz frequency and its delayed clocks are used as inputs. Besides, the outputs of digital parts of postsynaptic neurons are also conveyed to this module. If there is a firing postsynaptic neuron, the signal Sign goes high immediately.

The transient simulations of digital parts of the proposed neuron circuits during unsupervised learning are shown in Figure 12. During unsupervised learning, there is a fair competition among postsynaptic neurons, labels have no effect. The first postsynaptic neuron that makes a firing request becomes the

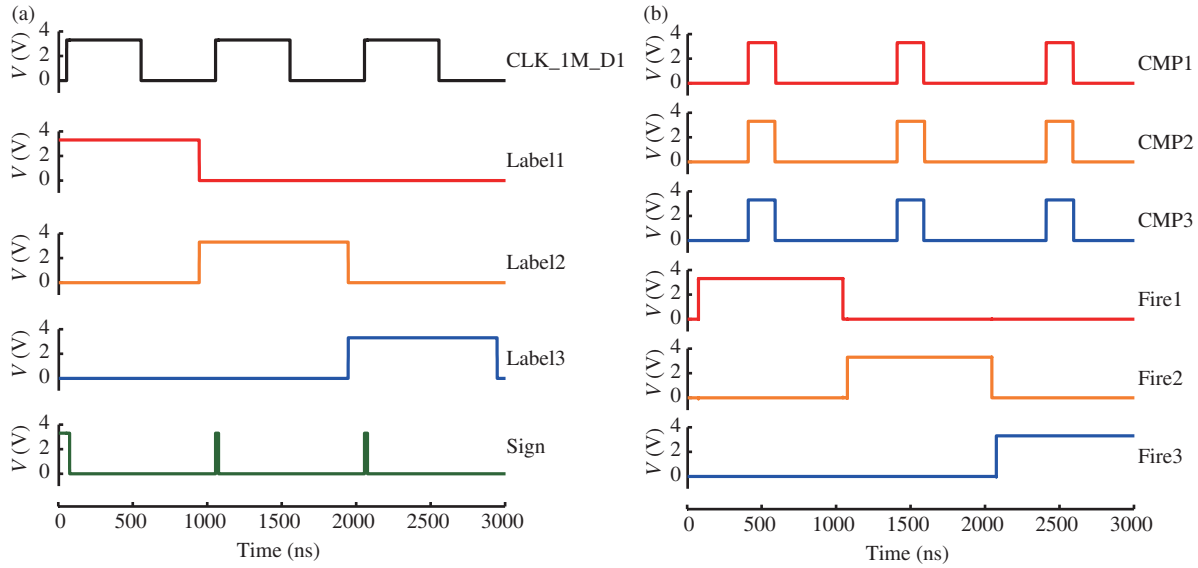

Figure 13 (Color online) The SPICE transient simulations of digital parts of the proposed neuron circuits during supervised learning. Alongside with each example, a label should be input to force one of the postsynaptic neurons to fire. (a) Waveforms of CLK_1M_D1, Label1, Label2, Label3, and Sign; (b) waveforms of CMP1, CMP2, CMP3, Fire1, Fire2, and Fire3.

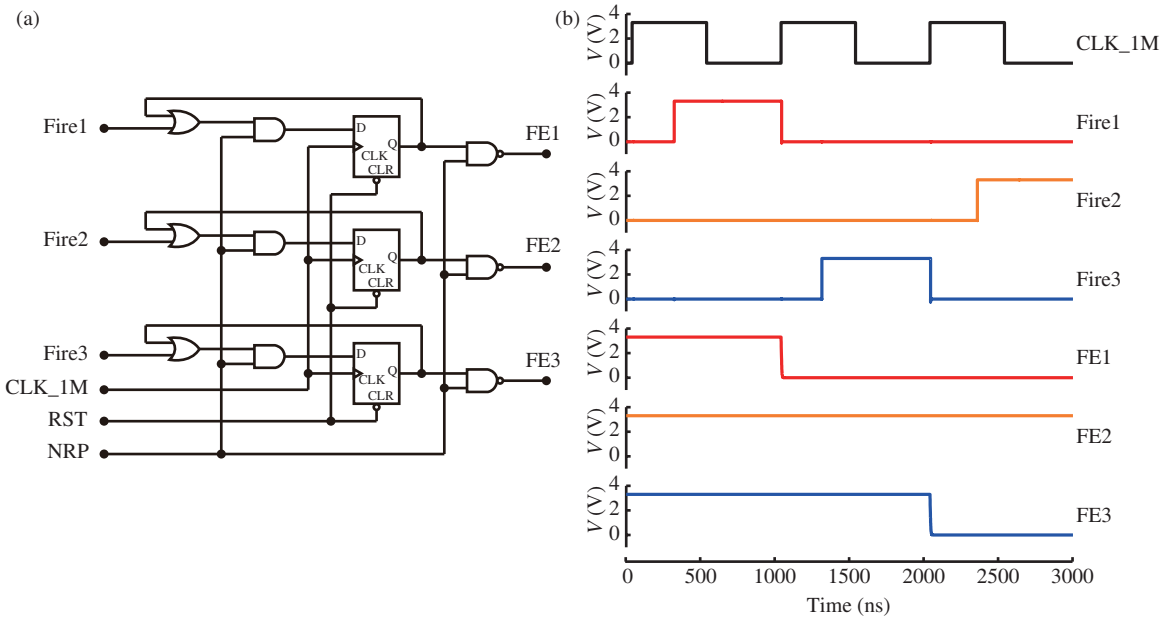

Figure 14 (Color online) (a) Circuit design of an NRP controller. (b) The SPICE transient simulations of the NRP controller. The NRP selection and the firing of a neuron itself control whether the neuron is allowed to fire during the subsequent operations together.

winner and fires, while others do not.

The transient simulations of digital parts of the proposed neuron circuits during supervised learning are shown in Figure 13. Alongside with each example, a label is input. Then the corresponding postsynaptic neuron fires, regardless of the firing requests.

4.4 NRP controller

The NRP controller implements the NRP functions of postsynaptic neurons, if necessary. The circuit design of the NRP controller is shown in Figure 14(a). The NRP controller accepts firing signals Fire1,

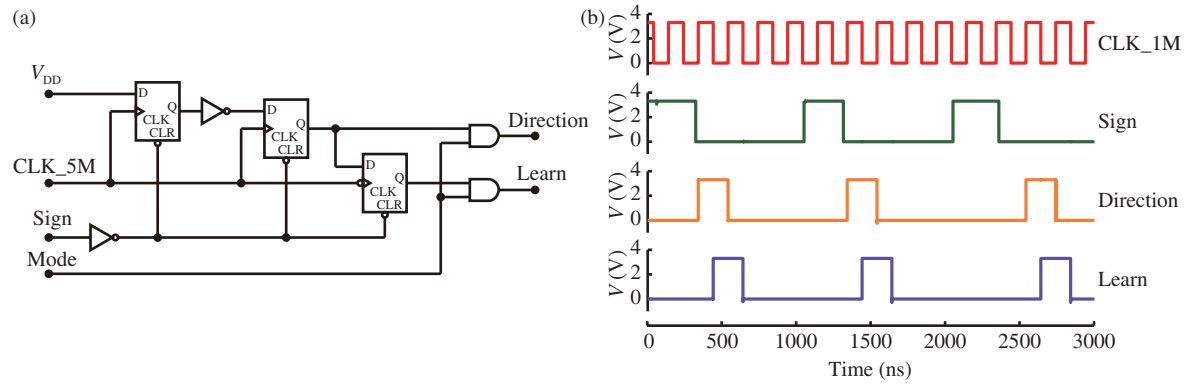

Figure 15 (Color online) (a) Circuit design of a learning controller. (b) The SPICE transient simulations of the learning controller. The learning controller generates two signals that are crucial for learning. The pulse of Learning immediately follows the pulse of direction, and the time difference is fixed to 100 ns.

Fire2 and Fire3 from the digital parts of postsynaptic neurons, and outputs firing enable signals FE1, FE2, and FE3. When the firing enable signal is high, the corresponding postsynaptic neuron is allowed to fire; otherwise, when the firing enable signal is low, the neuron is not allowed to fire.

The transient simulations of the proposed NRP controller are shown in Figure 14(b). It can be observed that after the firing of each neuron, its firing enable signal goes low, indicating that the neuron enters it NRP and is not allowed to fire again.

4.5 Learning controller

As shown in Figure 15(a), learning controller is designed to generate two signals that are crucial for learning: direction and learning. It has been mentioned above that both LTD and LTP are needed in the learning mode, direction is the signal that controls whether LTD or LTP can be applied. When direction is low, LTP can be applied; when direction is high, LTD can be applied. Nevertheless, the execution of LTD and LTP depends on the signal learning. When learning is low, neither LTD nor LTP is applied to the synapses. In this module, CLK_5M is chosen as the clock signal, since it results in a proper pulse width for RRAM operations.

The transient simulations of the proposed learning controller are shown in Figure 15(b). The pulse of learning immediately follows the pulse of direction, and the time difference is fixed to 100 ns. direction and learning control the learning operations together.

5 Systematic simulations and analysis

To verify the potential of the proposed learning method to handle complex classification tasks, systematic simulations are performed on MATLAB.

The MNIST database is used to test such a neuromorphic system. Since the presynaptic neurons only accept binary data, the MNIST database must be binarized first. Pixel values that are greater than 0.5 are regarded as "1", while the others are regarded as "0".

Then the 60000 examples in the training set of MNIST are used to train the system. Layer1 of the system performs unsupervised learning and Layer2 performs supervised learning. Examples are conveyed to the presynaptic neurons in Layer 1, and labels are conveyed to the postsynaptic neurons in Layer2.

SPICE simulations are performed to reveal the changes of RRAM synapses during learning. Figure 16 shows the resistance matrices of part of the RRAM arrays before and after learning. Before learning, the initial states of RRAM synapses are arbitrary; after learning, information is store into the RRAM synapses, which also indicates the memory of the example is created in the system.

After learning the whole MNIST database, the system is capable of classifying the handwritten digits. The system is tested by the 10000 examples in the testing set of MNIST. The capability relates to the

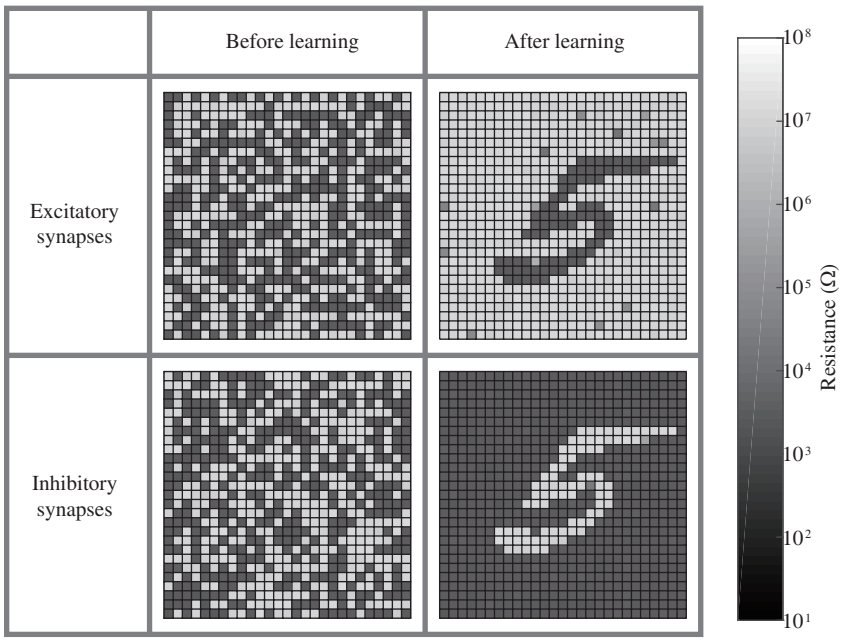

Figure 16 Resistance matrices of part of the RRAM arrays before and after learning. Before learning, the initial states of RRAM synapses are arbitrary; after learning, information is stored into the RRAM synapses, which also indicates the memory of the example is created in the system. Excitatory synapses and inhibitory synapses show opposite evolutions.

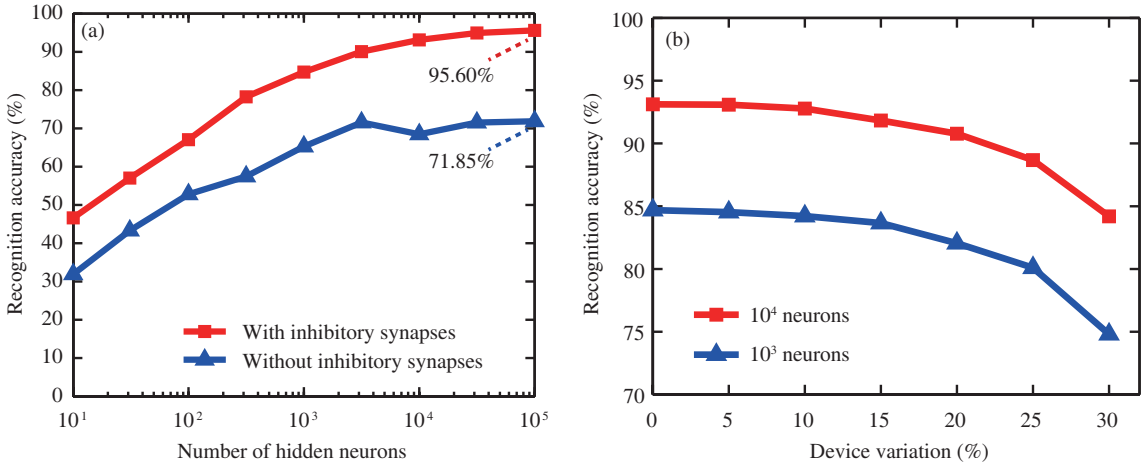

Figure 17 (Color online) (a) Recognition accuracy as a function of the number of hidden neurons. When inhibitory synapses are used, the system achieves a better recognition accuracy (up to 95.6%). (b) Recognition accuracy as a function of device variations. Although the recognition accuracy of the system decreases as the device variations increase, the system shows a relatively good tolerance. When 10 thousand neurons are used, the system can maintain a high recognition accuracy ($> 90\%$) when the device variation is up to 20%.

scale of the system. Figure 17(a) shows a relationship between the recognition accuracy and the number of hidden neu rons. It can be observed that the system can achieve a higher recognition accuracy with more hidden neurons. With 105 hidden neurons, the system achieves a peak recognition accuracy of 95.60%. Even with fewer hidden neurons, the results are still acceptable. Figure 17(a) also illustrates the necessity of inhibitory synapses. Without inhibitory synapses, the recognition accuracy of the system is at most 71.85%.

Besides, the device variation of RRAM is a crucial problem that has to be considered. We analyze the impact of device variations on the system performance. In this paper, the device variation is defined as the ratio of the standard deviation of device resistance and mean resistance, and the device resistance is supposed to have a normal distribution. The simulation results are shown in Figure 17(b). It can

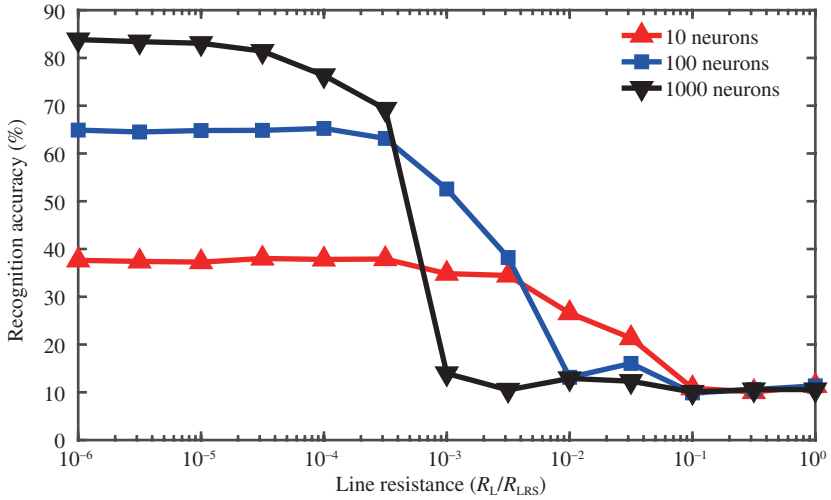

Figure 18 (Color online) The impact of line resistance on the recognition accuracy. RL is the line resistance between two adjacent RRAM cells, and RLRS is the average resistance of LRS RRAM cells. Systems with more hidden neurons are more sensitive to the IR drop issue.

Table 2 Resource utilization of digital circuits of the system with differenct number of hidden neurons

Hidden neurons	Logic utilization (in ALMs)	Total registers
10	88	913
100	441	1273
1000	4188	4861

be observed that although the recognition accuracy of the system decreases as the device variations increase, there is a good tolerance. If 10^4 hidden neurons are integrated in the system, the system is able to maintain a high recognition accuracy ($> 90\%$) even when the device variation is up to 20%. The results indicate that the proposed modified Hebbian learning is practical and suitable for RRAM-based neuromorphic systems.

The impact of line resistance in the array on the recognition accuracy is shown in Figure 18. The recognition accuracy of the system declines with the increase of line resistance. The line resistance causes IR drop issues in both the learning mode and the classification mode, and thus causes writing failures and deviations of output currents, which affect the accuracy. From Figure 18, we can see that systems with more hidden neurons are more sensitive to the IR drop issue.

If a learning method is complex, it generally brings difficulties to the hardware implementation and results in a complexity of control circuits. By analyzing the resource utilization of the system, we show that the proposed learning method is viable on hardware. Table 2 shows the resource utilization of digital circuits of the system. Here we use the hardware overhead on a field programmable gate array (FPGA) as a reference. The logic utilization is represented by the utilization of adaptive logical modules (ALMs). The proposed learning method is suitable for binary RRAM devices, thus it loosens the restriction of devices. By using transistor-free crossbar arrays, the system can show better scalability. Owing to the advantages of the proposed learning method, the system takes less time to learn each example, and achieves a higher recognition accuracy. More importantly, circuits for neuromorphic systems based on the proposed learning method have been developed and verified.

6 Conclusion

In this paper, a circuit design of RRAM-based neuromorphic hardware systems for classification and modified Hebbian learning is presented. Metal-oxide RRAM arrays with transistor-free crossbar structure are used as binary electronic synapses. The use of binary synapses can loosen the restriction of devices

and increase the feasibility of system implementation. With using the proposed learning method, the neuromorphic system can achieve a high learning efficiency. Simulation results show that the circuits are able to classify or learn 1 million examples within a second. Systematic simulations show that the neuromorphic system can achieve a maximum recognition accuracy of 95.6% towards the MNIST database, which also suggests the potential of RRAM-based neuromorphic systems in relatively complex tasks. In addition, a good tolerance to device variations is also obtained. The proposed modified Hebbian learning method is proved to be suitable for RRAM-based neuromorphic systems. Concrete circuit design and verification prove the feasibility of implementing such a learning method on hardware that is completely independent of the von Neumann architecture.

Taking advantage of the compatibility of metal-oxide RRAM with silicon CMOS technology, in the future, we may be able to integrate RRAM synapses with the CMOS-based neuron circuits on a single chip, and construct high-density and high-performance neuromorphic systems. Nevertheless, a tradeoff between system size and performance is still required.

This work may provide useful reference for the further implementation of RRAM-based neuromorphic hardware systems.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61421005, 61334007, 61604005).

References

1 Indiveri G, Chicca E, Douglas R. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. IEEE Trans Neural Netw, 2006, 17: 211–221
2 Mead C. Neuromorphic electronic systems. Proc IEEE, 1990, 78: 1629–1636
3 Yu S, Wu Y, Jeyasingh R, et al. An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. IEEE Trans Electron Device, 2011, 58: 2729–2737
4 Wong H S P, Lee H Y, Yu S, et al. Metal-oxide RRAM. Proc IEEE, 2012, 100: 1951–1970
5 Yang J J, Strukov D B, Stewart D R. Memristive devices for computing. Nat Nanotech, 2013, 8: 13–24
6 Yu S M, Gao B, Fang Z, et al. A neuromorphic visual system using RRAM synaptic devices with Sub-pJ energy and tolerance to variability: experimental characterization and large-scale modeling. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), San Francisco, 2012
7 Gao B, Bi Y, Chen H Y, et al. Ultra-low-energy three-dimensional oxide-based electronic synapses for implementation of robust high-accuracy neuromorphic computation systems. ACS Nano, 2014, 8: 6998–7004
8 Yu S, Li Z, Chen P Y, et al. Binary neural network with 16 Mb RRAM macro chip for classification and online training. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), San Francisco, 2016
9 Jang J W, Park S, Burr G W, et al. Optimization of conductance change in $Pr_{1-x}Ca_xMnO_3$-based synaptic devices for neuromorphic systems. IEEE Electron Device Lett, 2015, 36: 457–459
10 Prezioso M, Merrikh-Bayat F, Hoskins B D, et al. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. Nature, 2015, 521: 61–64
11 Soltiz M, Merkel C, Kudithipudi D, et al. RRAM-based adaptive neural logic block for implementing non-linearly separable functions in a single layer. In: Proceedings of IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), Amsterdam, 2012. 218–225
12 Ambrogio S, Balatti S, Milo V, et al. Novel RRAM-enabled 1T1R synapse capable of low-power STDP via burst-mode communication and real-time unsupervised machine learning. In: Proceedings of IEEE Symposium on VLSI Technology, Honolulu, 2016
13 Chu M, Kim B, Park S, et al. Neuromorphic hardware system for visual pattern recognition with memristor array and CMOS neuron. IEEE Trans Ind Electron, 2015, 62: 2410–2419
14 Milo V, Pedretti G, Carboni R, et al. Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), San Francisco, 2016
15 Cantley K D, Subramaniam A, Stiegler H J, et al. Hebbian learning in spiking neural networks with nanocrystalline silicon TFTs and memristive synapses. IEEE Trans Nanotechnol, 2011, 10: 1066–1073
16 Burr G W, Shelby R M, Sebastian A, et al. Neuromorphic computing using non-volatile memory. Adv Phys-X, 2017, 2: 89–124
17 Zhang Q T, Wu H Q, Yao P, et al. Sign backpropagation: an on-chip learning algorithm for analog RRAM neuromorphic computing systems. Neural Netw, 2018, 108: 217–223
18 Liao Y, Deng N, Wu H Q, et al. Weighted synapses without carry operations for RRAM-based neuromorphic systems. Front Neurosci, 2018, 12: 167
19 Lynch M A. Long-term potentiation and memory. Physiol Rev, 2004, 84: 87–136
20 Massey P V, Bashir Z I. Long-term depression: multiple forms and implications for brain function. Trends Neurosci, 2007, 30: 176–184

21 Diehl P U, Matthew C. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Front Comput Neurosci, 2015, 9: 99
22 Haykin S. The human brain. In: Neural Networks and Learning Machines. 3rd ed. New Jersey: Pearson Education, 2009
23 Chen A. A comprehensive crossbar array model with solutions for line resistance and nonlinear device characteristics. IEEE Trans Electron Device, 2013, 60: 1318–1326
24 Yu S, Chen P Y, Cao Y, et al. Scaling-up resistive synaptic arrays for neuro-inspired architecture: challenges and prospect. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), San Francisco, 2016
25 Hebb D O. The organization of behavior: a neuropsychological theory. In: the Organization of Behavior: A Neuropsychological Theory. New York: Wiley, 2013
26 Lowel S, Singer W. Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. Science, 1992, 255: 209–212
27 Haykin S. Principles of self-organization. In: Neural Networks and Learning Machines. 3rd ed. New Jersey: Pearson Education, 2009
28 Stent G S. A physiological mechanism for Hebb's postulate of learning. Proc Natl Acad Sci USA, 1973, 70: 997–1001
29 Changeux J P, Danchin A. Selective stabilisation of developing synapses as a mechanism for the specification of neuronal networks. Nature, 1976, 264: 705–712
30 Principe J C, Euliano N R, Lefebvre W C. Hebbian learning and principal component analysis. In: Neural and Adaptive Systems: Fundamentals Through Simulations. 2nd ed. New York: Wiley, 1997