Energy Efficient Spiking Neural Network Design with RRAM Devices

Tianqi Tang[1], Rong Luo[1], Boxun Li[1], Hai Li[2], Yu Wang[1], Huazhong Yang[1]
[1]Dept. of E.E., Tsinghua National Laboratory for Information Science and Technology (TNList),
Tsinghua University, Beijing, China
[2]Dept. of E.C.E., University of Pittsburgh, Pittsburgh, USA
[1] Email: yu-wang@mail.tsinghua.edu.cn

***Abstract*—The brain-inspired neural networks have demonstrated great potential in big data analysis. The spiking neural network (SNN), which encodes the real world data into spike trains, promises great performance in computational ability and energy efficiency. Moreover, it is much more biologically plausible than the traditional artificial neural network (ANN), which keeps the input data in its original form. In this paper, we introduce an RRAM-based energy efficient implementation of STDP-based spiking neural network cascaded with ANN classifier. The recognition accuracy and power consumption are compared between SNN and traditional three-layer ANN. The experiments on the MNIST database demonstrate that the proposed RRAM-based spiking neural network requires only 14% of power consumption compared with RRAM-based artificial neural network with a slight accuracy decay (∼2%).**

I. Introduction

The explosion of "big data" applications promotes huge demands for higher processing speed, lower power consumption, and better scalability of computing systems. However, the classic "scaling down" method approaches to the end, making it difficult for traditional CMOS-based computing systems to achieve considerable performance improvements [1]. Moreover, from the architecture level, the traditional von Neumann architecture faces the "memory wall" bottleneck. The speed-up of memory access time cannot catch up with that of processor frequency [2]. New technologies, from both the device level and the architecture level, are required to overcome these challenges.

The spiking neural network (SNN), shown in Fig. 1 is a bio-inspired model abstracted from actual neural system, which encodes and processes information with spikes [3]. Compared with the traditional von Neumann architecture, SNN combines the computation and memory together and breaks through the 'memory wall' bottleneck. Moreover, the similarity between the spiking neural network and the brain makes SNN a promising tool to deal with cognitive tasks, ranging from the image classification to the speech recognition [4], [5].

However, the spiking neural network faces a severe problem of hardware implementation efficiency. For example, IBM made the cat cortex simulation (with $\sim 10^9$ neurons and $\sim 10^{13}$ synapses) on Blue Gene supercomputer cluster (with 147,456 CPUs and 144TB memory). The power consumption reached 1.4MW, which was of five orders of magnitude higher than the human brain (∼ 20W) [6]. There is a huge demand for the energy efficient implementation of spiking neural networks.

The emerging RRAM devices demonstrate a promising solution. The ultra-high integration density of RRAM enables a large number of signal connections within a small circuit size [7] [8]. More importantly, the RRAM crossbar array can naturally realize the matrix-vector multiplication efficiently, which is one of the most important operations in spiking neural networks [5], [9]. A lot of work has explored the potential of computing with RRAM crossbar array. For example, a low power on-chip neural approximate computing system has been proposed with power efficiency of more than 400 GFLOPS/W [8] [10].

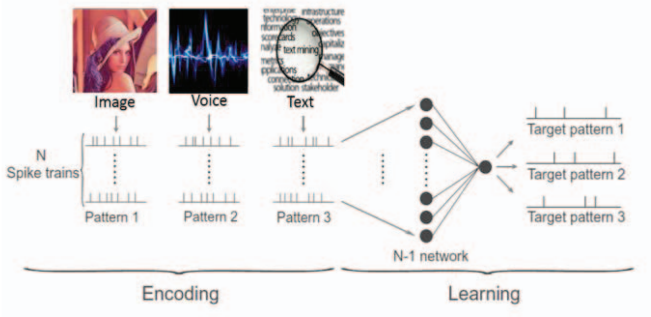


Fig. 1. Spiking Neural Network

In this work, we implement an energy efficient spiking neural network with emerging RRAM devices. The contribution of this paper includes:

1) We implemented a five-layer spiking neural network framework: The first two layers are made up of spiking neurons and the synaptic matrix learns according to spiking-time-dependent-plasticity (STDP) weight updating rule; the last three layers are made up of a three-layer artificial neural network classifier. Moreover, there is a spike decoding module between them.
2) We compared the recognition accuracy and power consumption with that of an RRAM-based traditional artificial neural network. Experiment results show that the SNN system achieves only ∼2% recognition accuracy decay with only ∼14% power consumption on MNIST dataset.

The rest of this paper is organized as follows: Section II introduces the related background knowledge and Section III proposes the RRAM-based spiking neural network. A case study of digit recognition tasks is introduced in Section IV to evaluate the performance of RRAM-based SNN followed by the conclusion in Section V.

II. Preliminaries

A. Spike Neurons and Synapses

The neuron is the basic building block of SNN. Different mathematical models of spiking neurons have been explored with different levels of computational efficiency and biological plausibility [11]. The model of *Leaky Integrate and Fire (LIF)* [12] is one of the most widely used models for its computing efficiency. In this model, a one-order differential function determines the state variable $V(t)$ and a

978-1-4799-4833-8/14/$31.00 ©2014 IEEE

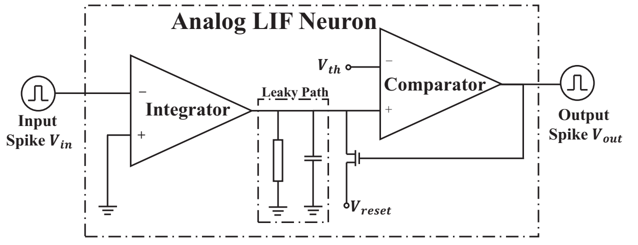


Fig. 2. Analog LIF Neuron

threshold function determines whether the neuron spikes and then resets. And it is described as:

$$V(t) = \begin{cases} \beta \cdot V(t-1) + V_{in}(t) & \text{when } V < V_{th} \\ V_{reset} \text{ and set a spike} & \text{when } V \geq V_{th} \end{cases} \quad (1)$$

where $V(t)$ is the state variable and β is the leaky parameter; V_{th} is the threshold state which the state variable makes comparison with and once exceeding, the state variable will reset to V_{reset}.

An analog *LIF* neuron implementation is shown as Fig. 2: the integrator calculates the state of the neuron $V(t)$ and the RC works as the leaky path. When $V(t) > V_{th}$, the transistor will be conducted and $V(t)$ will be reset.

B. Synapse and Synaptic Weight Learning Rule: STDP

Synapses connect neurons to each other and transmit signals between them. The weight of synapses, which determines the connecting strength of neurons, are learnable. Spike Timing Dependent Plasticity (STDP) [13] is an unsupervised learning rule that updates the synaptic weights as a function of the relative spiking time of pre- and post-synaptic neurons and the exponential window form of STDP is shown as:

$$\Delta w = \begin{cases} a^{+} \cdot w_{ij}(1-w_{ij}) \cdot \exp(-\frac{|t_j - t_i|}{\tau}) & \text{if } t_j \geq t_i \\ a^{-} \cdot w_{ij}(1-w_{ij}) \cdot \exp(-\frac{|t_j - t_i|}{\tau}) & \text{if } t_j < t_i \end{cases} \quad (2)$$

where w_{ij} is the synaptic weight between pre- and post-synapse neuron n_i, n_j; t_i, t_j are the spiking time of neuron n_i, n_j; a is the maximum learning rate and τ is the time constant of the learning window. According to Eq. (2), the synaptic weight is limited in the interval of $[0, 1]$. The learning rate is decided by the time interval of n_i, n_j spiking: The closer between pre- and post-synaptic spikes, the larger the learning rate. The weight update direction is decided by which neuron spikes first: For the excitatory neuron, if the post-synaptic neuron n_j spikes later than n_i, the synapse will be strengthened; otherwise, it will be decayed; for the inhibitory neuron, vice versa. When every synaptic weight no longer changes or is set to 0/1, the learning process is finished.

C. RRAM Device Characteristics

Fig. 3(a) shows a 1D filament model of HfO$_x$ based RRAM device [9]. The model is a sandwich structure with a resistive layer between two metal electrodes. The conductance is exponentially dependent on the tunneling gap (d). Therefore, we will take advantage of the variable conductance of the RRAM device by setting the value of tunneling gap d. For the HfO$_x$ based RRAM device, the I-V relationship can be empirically expressed as follows [9]:

$$I = I_0 \cdot \exp(-\frac{d}{d_0}) \cdot \sinh(\frac{V}{V_0}) \quad (3)$$

where d is the average tunneling gap distance. I_0 ($\sim 1mA$), d_0 ($\sim 0.25nm$) and V_0 ($\sim 0.25V$) are fitting parameters through experiments. When $V << V_0$, there exists the approximation that $\sinh(\frac{V}{V_0}) \approx \frac{V}{V_0}$. The I-V relationship is linear under this condition. In this work, we will scale down the RRAM voltage to under 0.1V in order to take advantage of the approximately linear I-V relationship.

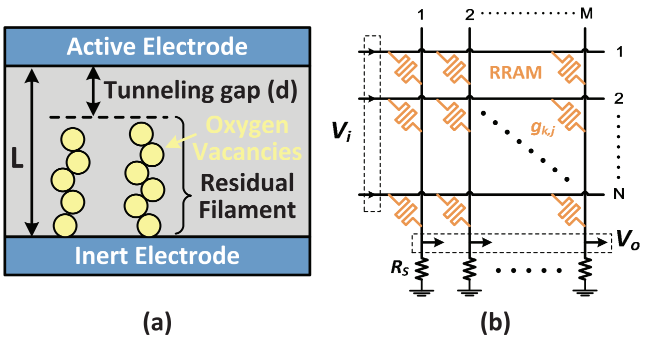


Fig. 3. (a). Physical model of the HfO$_x$ based RRAM. The resistance of the RRAM device is determined by the tunneling gap distance d, and d will evolve due to the filed and thermally driven oxygen ion migration. (b). Structure of the RRAM Crossbar Array.

III. Spiking Nerual Network Learning System

The system is a five-layer neural network system, with two-layer spiking based neural network and a three-layer artificial neural network. There is a converting module between them to convert the spiking trains into the spike count vectors. Then the spike count vectors are sent into the following layers of the network. The system scheme is shown in Fig. 4 and each module is introduced as follow:

A. Spike Encoding Module

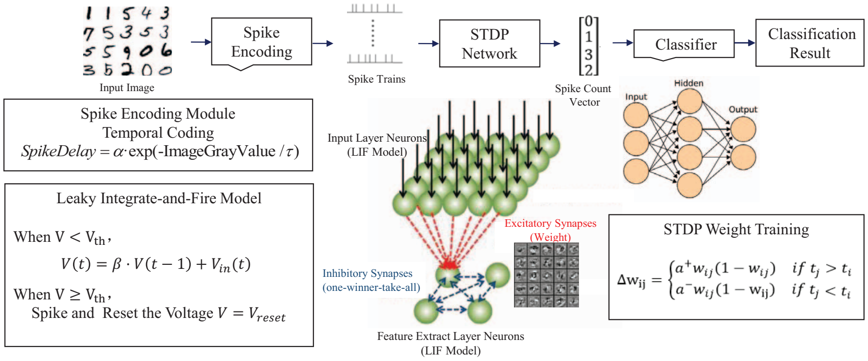
Since the spike propagates information in the spiking network, an encoding module is needed to encode the original data into spiking trains. In this work, the temporal coding [14] is introduced to transform the gray value of the image pixel into the pulse delay as shown in Eq. 4, where α is a fitting parameter and the coefficient *MAXGRAYVALUE* is used to scale the original gray value $ImageGrayValue$ from $[0, 255]$ to the interval of $[0, 1]$.

$$SpikeDelay = \alpha \cdot (1 - \frac{ImageGrayValue}{MAXGRAYVALUE}) \quad (4)$$

B. Two-Layer Spike based Neural Network

For the five-layer neural network system, the first two levels are made up of spiking neurons. The spiking patterns of input layer neurons are determined by the encoding; while in the feature representation layer, the neurons are all LIF neurons and they integrate the input spikes according to the synaptic matrix weight, spike and reset when the state voltages exceed the threshold. When training, the synaptic weight matrix W is being optimized according to the STDP learning rule, which tries to make the feature representation layer hold the information in the input layer and to make the information represented by each neuron in the representation layer different from each other [4].

Another thing that should be noted is the connection between the neurons of the same layer. For the input layer, each neuron is assumed to be independent from each other. However, for the representation layer, the neurons are inhibitory from each other – for one training cycle, if one LIF neuron fires, then all the other LIF neurons in the representation layers will be reset.



Input Image
Spike Encoding
Spike Trains
STDP Network
0 1 3 2
Spike Count Vector
Classifier
Classification Result
Spike Encoding Module
Temporal Coding
SpikeDelay = α·exp(-ImageGrayValue / τ)
Leaky Integrate-and-Fire Model
When V < V$_{th}$,
$V(t) = \beta \cdot V(t-1) + V_{in}(t)$
When V ≥ V$_{th}$,
Spike and Reset the Voltage $V = V_{reset}$
Input Layer Neurons (LIF Model)
Input
Hidden
Output
Excitatory Synapses (Weight)
Inhibitory Synapses (one-winner-take-all)
Feature Extract Layer Neurons (LIF Model)
STDP Weight Training
$\Delta w_{ij} = \begin{cases} a^+ w_{ij}(1-w_{ij}) & if\ t_j > t_i \\ a^- w_{ij}(1-w_{ij}) & if\ t_j < t_i \end{cases}$

Fig. 4. SNN+ANN System Scheme

C. Three-Layer Arificial Neural Network Classifier

A three-layer artificial neural network structure, working as a classifier, is cascaded after the spike based neural network module introduced in Section III-B. In this way, a complete recognition system is built where spike-based neural network works as the feature representation module and ANN works as the classifier. The neurons introduced here are sigmoid neurons which make an nonlinear mapping from input voltage value to output voltage value. The formulation is as below:

$$V_{out} = \frac{1}{1 + \exp\left(-V_{in}/V_0\right)} \tag{5}$$

D. Spike Decoding Module

Since the spike, which carries information through the delay of the pulse, runs in the first two layers of the neural network system and there is a three-layer classifier cascaded after that, a counter is introduced as the decoding module to link the SNN feature representation module and the ANN classifier. The input layer in the SNN module keeps on sending spikes into the network according to the *SpikeDelay* defined in Eq. 4 during the given time interval. Then the decoding module sends the spike count vector into the ANN module after scaling down the vector to [0, 1].

E. System Implementation on RRAM

The system is made up of neurons and inter-layer weight matrices. The spiking neuron of the first two layers can be implemented by the analog LIF neuron shown in Fig. 2 and the sigmoid neuron of the last three layers can be implemented as [15]. For any two connected layers, the relationship between the input voltage vector ($\vec{V_i}$) and output voltage vector ($\vec{V_o}$) can be expressed as a matrix-vector multiplication operation, which is shown as follow [16]:

$$\begin{bmatrix} V_{o,1} \\ \vdots \\ V_{o,M} \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{1N} \\ \vdots & \ddots & \vdots \\ c_{M1} & \cdots & c_{MN} \end{bmatrix} \begin{bmatrix} V_{i,1} \\ \vdots \\ V_{i,N} \end{bmatrix} \tag{6}$$

Meanwhile, as shown in Fig. 3(b), the matrix-vector multiplication operation can be implemented on the RRAM crossbar. Supposing that k (k = 1,2,..,N) and j (j = 1,2,..,M) are the index numbers of input and output voltages, the relationship between the input and output voltage can be shown as

$$V_{o,j} = \frac{\sum_{k=1}^{N} g_{jk} V_{i,k}}{g_s + \sum_{k=1}^{N} g_{jk}} \tag{7}$$

where g_{jk} and g_s are respectively the conductivity of the RRAM device and the load resistor.

For the crossbar in the artificial neural network classifier module, since the input voltage vector of one training/testing sample is constant , the crossbar power consumption can be calculated as:

$$P = \sum_{k=1}^{N} \left(V_{i,k} \cdot \sum_{j=1}^{M} g_{jk} (V_{i,k} - \frac{\sum_{l=1}^{N} g_{jl} V_{i,l}}{g_s + \sum_{l=1}^{N} g_{jl}}) \right) \tag{8}$$

However, for the crossbar in the spiking-based module, the input voltage of one training/testing sample is a time-variant vector of which every dimension is encoded into 0/1. Therefore, the average power consumption can be calculated as:

$$\bar{P} = \frac{1}{T} \sum_{t=0}^{T} \left(\sum_{k=1}^{N} \left(V_{i,k}(t) \cdot \sum_{j=1}^{M} g_{jk} (V_{i,k}(t) - \frac{\sum_{l=1}^{N} g_{jl} V_{i,l}(t)}{g_s + \sum_{l=1}^{N} g_{jl}}) \right) \right) \tag{9}$$

According to Eq. (7), the matrix parameter c_{jk} can be represented by the conductivity of the RRAM device (g_{jk}) and that of the load resistors (g_s) as:

$$c_{jk} = \frac{g_{jk}}{g_s + \sum_{l=1}^{N} g_{jl}} \tag{10}$$

Therefore, there does not exist a direct one-to-one mapping from the original weight matrix C to the crossbar conductance matrix G. Moreover, some physical limitation on G should be considered:

1) The item c_{jk} of the original weight matrix C can be either positive or negative while every item the conductance of RRAM crossbar G should be positive. Thus, the original weight matrix C should be decomposed into two parts: one positive C^+, the other negative C^-;
2) The RRAM conductance g_{jk} has the physical limitation that $g_{min} \leq g_{jk} \leq g_{max}$. For simplicity, we would like to make a linear mapping:

$$g_{jk} = c_{jk} \cdot (g_{max} - g_{min}) + g_{min} \tag{11}$$

As shown in [16], there exists the relationship $c_{jk} \propto g_{max}/g_s$ when $g_s \gg g_{max}$.

IV. A Case Study: MNIST Dataset

In this section, a case study is made on MNIST digit recognition dataset to evaluate the performance of the five-layer spiking neural network system framework proposed in Section III.

A. Experiment Setup

In this work, the training process is implemented on the CPU platform where LIF neurons in Eq. (1) are used in the first two layers and the sigmoid neurons are used in the last three layers. For

TABLE I
EXPERIMENT RESULTS OF SNN+ANN SYSTEM WITH DIFFERENT NETWORK SIZES.
KEY PARAMETERS: SNN INPUT VOLTAGE: 0.1V, ANN INPUT VOLTAGE: 0.9V, LOAD RESISTANCE: $10k\Omega$

Network Size	Accuracy on CPU(%)	Accuracy on RRAM(%)	Power on RRAM(mW)
784×10 SNN+10×50×10 ANN	67.8	65	37.23
784×50 SNN+50×50×10 ANN	91.7	88	186.18
784×100 SNN+100×50×10 ANN	91.5	90	327.36
784×100×10 ANN	94.3	92	2273.60

the testing process, we make the circuit simulation where the weight matrix is mapped to RRAM-based crossbar introduced in Section III-E.

The MNIST dataset is used to test the performance of RRAM-based neural network system proposed in Section III. MNIST is a widely used dataset for optical character recognition with 60,000 handwritten digits in training set and 10,000 for testing set. In our experiment, we use all the examples of handwritten digits of 0∼9 to train the neural network system and randomly select 1,000 samples for testing.

Since the input images are 28×28 sized 256-level gray images. The five-layer spiking neural network system has five layers of neurons in all and experiments are made with different network sizes of $784\times\{10{,}50{,}100\}\times50\times10$ where the variable is the neuron number of feature representation layer.

B. Performance of the Framework

The simulation results summarized in Table I show that the computation accuracy obtained by the spiking neural network system is compatible to that of traditional three-layer artificial neural network structure when setting the proper network size. The best performance of the spiking neural network system achieves only ∼2% recognition accuracy degradation on RRAM platform. For the power consumption, the RRAM-based spiking neural network requires only 14% of that on artificial neural network. The reasons for lower power consumption of SNN than that of ANN are listed as follow:

1) For the first two layers of the SNN (the feature representation module), the input voltage can be binary since it transforms the numerical information into the temporal domain. Therefore, there is no need for SNN to hold a large voltage range to represent multiple input states as implemented in ANN.
2) For the last three layers of the SNN (the ANN classifier module), the network scale is much smaller than that of ANN in our experiment. Therefore, the power consumption is much less.

V. CONCLUSION

In this work, we implement an energy efficient spiking neural network with emerging RRAM devices and make a comparison on performance and power consumption with the spiking neural network and traditional neural network. Experiment results on MNIST dataset show that the spiking neural network achieves only ∼2% recognition accuracy decay with only 14% power consumption on RRAM platform under ideal condition (without considering interconnection effect, nonlinear effect, etc).

However, there are still many challenges remaining in this spiking neural network structure. For example, the encoding mechanism from original data to spiking is not quite clear. It perhaps has a huge effect on system performance and power efficiency. Thus, how to design an a proper encoding mechanism is one possible method of improving the performance of the system. In addition, the non-ideal circuit condition (e.g. the interconnection effect, the input variation) should be discussed in the future work.

ACKNOWLEDGMENTS

This work was supported by 973 project 2013CB329000, National Science and Technology Major Project (2011ZX03003-003-01, 2013ZX03003013-003) and National Natural Science Foundation of China (No.61373026, 61261160501, 61271269), and Tsinghua University Initiative Scientific Research Program. And we gratefully acknowledge the support from Prof. Shimeng Yu with the help of RRAM model.

REFERENCES

[1] L. Chang, Y.-k. Choi, D. Ha, P. Ranade, S. Xiong, J. Bokor, C. Hu, and T.-J. King, “Extremely scaled silicon nano-cmos devices,” *Proceedings of the IEEE*, vol. 91, no. 11, pp. 1860–1873, 2003.

[2] W. A. Wulf and S. A. McKee, “Hitting the memory wall: implications of the obvious,” *ACM SIGARCH computer architecture news*, vol. 23, no. 1, pp. 20–24, 1995.

[3] W. Maass, “Networks of spiking neurons: the third generation of neural network models,” *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[4] T. Masquelier and S. J. Thorpe, “Unsupervised learning of visual features through spike timing dependent plasticity,” *PLoS computational biology*, vol. 3, no. 2, p. e31, 2007.

[5] D. Querlioz, W. Zhao, P. Dollfus, J. Klein, O. Bichler, and C. Gamrat, “Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches,” in *Nanoscale Architectures (NANOARCH), 2012 IEEE/ACM International Symposium on*. IEEE, 2012, pp. 203–210.

[6] D. Kuzum, R. G. Jeyasingh, B. Lee, and H.-S. P. Wong, “Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing,” *Nano letters*, vol. 12, no. 5, pp. 2179–2186, 2011.

[7] D. Niu, C. Xu, N. Muralimanohar, N. P. Jouppi, and Y. Xie, “Design of cross-point metal-oxide reram emphasizing reliability and cost,” in *Computer-Aided Design (ICCAD), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 17–23.

[8] J. J. Yang, D. B. Strukov, and D. R. Stewart, “Memristive devices for computing,” *Nature nanotechnology*, vol. 8, no. 1, pp. 13–24, 2013.

[9] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H.-S. P. Wong, “Stochastic learning in oxide binary synaptic device for neuromorphic computing,” *Frontiers in neuroscience*, vol. 7, 2013.

[10] B. Li, Y. Shan, M. Hu, Y. Wang, Y. Chen, and H. Yang, “Memristor-based approximated computation,” in *Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 242–247.

[11] E. M. Izhikevich, “Which model to use for cortical spiking neurons?” *IEEE transactions on neural networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[12] G. Indiveri, “A low-power adaptive integrate-and-fire neuron circuit,” in *ISCAS (4)*, 2003, pp. 820–823.

[13] S. Song, K. D. Miller, and L. F. Abbott, “Competitive hebbian learning through spike-timing-dependent synaptic plasticity,” *Nature neuroscience*, vol. 3, no. 9, pp. 919–926, 2000.

[14] P. Dayan and L. Abbott, “Theoretical neuroscience: computational and mathematical modeling of neural systems,” *Journal of Cognitive Neuroscience*, vol. 15, no. 1, pp. 154–155, 2003.

[15] K. Doya and S. Yoshizawa, “Memorizing oscillatory patterns in the analog neuron network,” in *Neural Networks, 1989. IJCNN., International Joint Conference on*. IEEE, 1989, pp. 27–32.

[16] M. Hu *et al.*, “Hardware realization of bsb recall function using memristor crossbar arrays,” in *DAC*. ACM, 2012, pp. 498–503.