

# An Overview on Memristor Crossbar Based Neuromorphic Circuit and Architecture

(Invited Paper)

Zheng Li, Chenchen Liu, Yandan Wang, Bonan Yan, Chaofei Yang, Jianlei Yang, and Hai (Helen) Li  
 Department of Electrical and Computer Engineering  
 University of Pittsburgh, Pittsburgh, PA 15261  
 Email: <http://www.ei-lab.org>

**Abstract**—As technology advances, artificial intelligence becomes pervasive in society and ubiquitous in our lives, which stimulates the desire for embedded-everywhere and human-centric intelligent computation paradigm. However, conventional instruction-based computer architecture was designed for algorithmic and exact calculations. It is not suitable for handling the applications of machine learning and neural networks that usually involve a large sets of noisy and incomplete natural data. Instead, neuromorphic systems inspired by the working mechanism of human brains create promising potential. Neuromorphic systems possess a massively parallel architecture with closely coupled memory and computing. Moreover, through the sparse utilizations of hardware resources in time and space, extremely high power efficiency can be achieved. In recent years, the use of memristor technology in neuromorphic systems has attracted growing attention for its distinctive properties, such as nonvolatility, reconfigurability, and analog processing capability. In this paper, we summarize the research efforts in the development of memristor crossbar based neuromorphic design from the perspectives of device modeling, circuit, architecture, and design automation.

**Keywords**—Neuromorphic computing, neuromorphic circuit and architecture, memristor, crossbar array, resistive memory.

## I. INTRODUCTION

The demand of high performance computing continuously increases as artificial intelligence becomes pervasive in society and ubiquitous in our lives. However, traditional von Neumann computer architecture designed for algorithmic and exact calculations becomes less efficient and scalable. Neuromorphic hardware systems inspired by the working mechanism of human brains [1] potentially can provide the capabilities of biological perception and cognitive information processing within a compact and energy-efficient platform. Therefore, the development of neuromorphic systems has gained a great deal of attention in recent years. Besides conventional CPUs, GPUs, or FPGAs [2][3], the use of emerging technologies such as resistive memory devices (a.k.a. memristor) in neuromorphic design has also been studied [4][5].

As early as in 1971, Professor Chua predicted the existence of memristor based on circuit theory [6]. Forty years later in 2008, the physical realization of memristor was demonstrated through a  $\text{TiO}_2$  thin-film [7]. Afterwards, many memristive materials and devices have been rediscovered [8]. A memristor can record its total electrical flux as memristance ( $M$ ). The feature is highly similar to weighting function of a biological synapse. Moreover, the two-terminal thin-film device

structure can be easily integrated into crossbar arrays. It can provide a large number of signal connections within a small footprint and conduct the weighted combination of input signals, making it very promising for massively-parallel, large-scale neuromorphic systems [9].

In this paper, we give an overview on the research efforts in developing neuromorphic circuit and architecture design that leverage memristor crossbar structure. A comprehensive view including device modeling, circuit, architecture, and design automation will be covered in the following sections.

## II. MEMRISTOR DEVICE MODELING

Fig. 1(a) illustrates the memristor device realized on a Pt/ $\text{TiO}_2$ /Pt thin-film structure [7]. The memristive function is achieved through the doping front movement, which can be controlled by external voltage excitation. And its overall memristance is determined by the ratio of the stoichiometric  $\text{TiO}_2$  with low conductivity and the semiconductor-like oxygen-deficient titanium dioxide ( $\text{TiO}_{2-x}$ ). Thus, it can be modeled as a coupled variable resistor model shown in Fig. 1(b), which is equivalent to two series-connected resistors such as

$$M(\alpha) = R_L \cdot \alpha + R_H \cdot (1 - \alpha). \quad (1)$$

Here  $\alpha$  ( $0 \leq \alpha \leq 1$ ) is the ratio of doping front position over the total thickness of  $\text{TiO}_2$  thin-film, represented by the relative doping front position. The velocity of doping front movement  $v(t)$ , which is driven by the voltage applied across the memristor  $V(t)$  can be expressed as

$$v(t) = \frac{d\alpha}{dt} = \mu_v \cdot \frac{R_L}{h^2} \cdot \frac{V(t)}{M(\alpha)}, \quad (2)$$

where  $\mu_v$  is the equivalent mobility of dopants,  $h$  is the total thickness of the  $\text{TiO}_2$  thin-film; and  $M(\alpha)$  is the total memristance which is a function of  $\alpha$ .

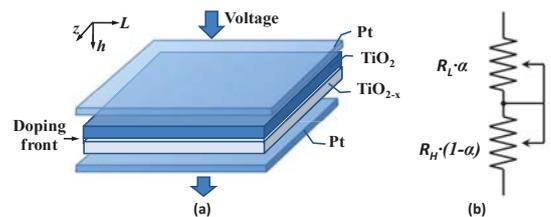


Fig. 1.  $\text{TiO}_2$  thin-film memristor. (a) structure, and (b) equivalent circuit.

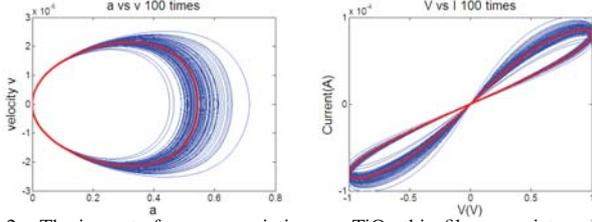


Fig. 2. The impact of process variations on  $\text{TiO}_2$  thin-film memristors. left:  $v$  vs.  $\alpha$ , right:  $I-V$  characteristics. (The blue curves are from 100 Monte-Carlo simulations, and red lines are the ideal condition.)

Note that the above bulk model is derived from the mathematical definition of memristor, which assumes a flat doping front moving up or down. In reality, however, filamentary conduction has been observed in nano-scale semiconductors: the current travels through some high conducting filaments rather than evenly passes the entire device [10]. Moreover, as process technology shrinks down, device parameter fluctuations incurred by process variations severely affecting the electrical characteristics. The situation in a memristor could be even worse: the parameter variations can result in the shift of electrical responses, which in turn affect the memristance since the total charge through a memristor indeed is the historic behavior of its current profile.

An approach to converge the difference between the bulk and filament models was to divide a  $\text{TiO}_2$  thin-film into many tiny filaments and adopt the bulk model to the small flat doping front in each filament [11]. The implications of memristor parameters to the circuit design was explored by taking into account the impact of memristor geometry variations. Fig. 2 shows the dynamic responses of 100 Monte Carlo simulations which can visually demonstrate the overall impact of process variations on the memristive behavior.

Moreover, metal oxide based memristor behaves stochastically and hence even a single memristive device demonstrates large variations in performance. More specific, the static states of a single memristor are not fixed, but have large variations with skewed distributions and heavy tails [12]. The switching mechanism of a memristor, that is, its dynamic behavior, performs as a stochastic process [13]. A stochastic behavior model which bypasses material-related parameters while directly linking the device analog behavior to stochastic

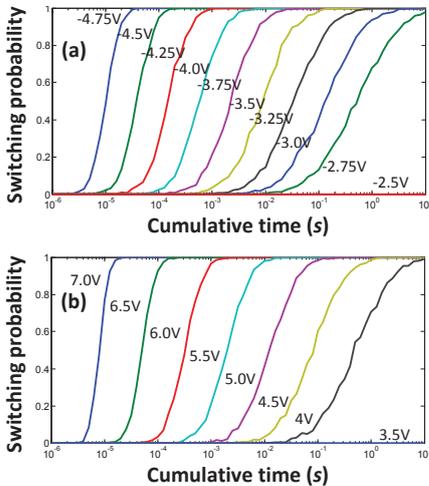


Fig. 3. The time dependency of ON (a) and OFF (b) switching at different external voltage  $V$ .

functions was presented to better facilitate the exploration of memristors in hardware implementation. Fig. 3 shows the time dependencies of ON and OFF switching probability at different applied voltages. The results have high approximation to the experimental results [13].

### III. NEUROMORPHIC CIRCUIT DESIGN

The applications of memristor crossbar in acceleration of scientific and neuromorphic computing have been studied. For example, the matrix-vector computation can be conducted through crossbar arrays by using voltage/current magnitudes to represent the data [5] or through a spiking neural network [14].

Fig. 4 depicts an overview of the spiking computing design that leverages the compact memristor crossbar structure [14]. It adopts the rate coding model and represents data using the frequency of spikes [15]. Through different bitlines (BLs) in the crossbar, the synaptic weighting functions of different entries are executed in parallel. The *integrate and fire circuits* (IFC) as post-neurons generate output spikes based on the strength of the weighted pre-neuron signals from the crossbar.

A single-layer neural network with  $N$  pre-neurons and  $M$  post-neurons can be implemented using a  $N \times M$  crossbar. First, the activity pattern of pre-neurons  $\mathbf{x}_{N \times 1}$  is transferred into a set of pulses to wordlines (WLs). The number of spikes on  $\text{WL}_i$  within the computation period ( $n_{x,i}$ ) corresponds to  $x_i \in \mathbf{x}$ . The weight from the  $j^{\text{th}}$  pre-neuron and the  $i^{\text{th}}$  post-neuron maps to the conductance  $g_{ij}$  at the crosspoint of  $\text{WL}_i$  and  $\text{BL}_j$ . The total weighted signal to post-neuron  $j$  is transferred to the current flowing through  $\text{BL}_j$  and accumulated on a capacitor  $C_m$  in IFC. Once the voltage on  $C_m$  reaches to a predefined threshold  $V_{th}$ , the IFC fires an output spike and resets  $C_m$ . The activity function of post-neurons  $\mathbf{y}_{M \times 1}$  is represented by a set of spike numbers such as  $[n_{y,0}, n_{y,1}, \dots, n_{y,M-1}]^T$ .

Under ideal condition without taking into account the realistic factors in circuit implementation, the spike number produced at the  $j^{\text{th}}$  post-neuron  $n_{y,j} \propto \sum_{i=0}^{N-1} g_{ij} \delta_i$ , where  $\delta_i$  corresponds the spike occurrence at  $\text{WL}_i$ . The assumption  $\sum_{i=0}^{N-1} g_{ij} \rightarrow 0$ , however, is satisfied only when all the resistive devices are at (or close to) the high resistance state. This cannot be generalized as a common condition in applications. Moreover, the delay overhead of IFC to generate pulses and reset  $C_m$  cannot be ignored.

The delay of IFC is a critical parameter determining the performance of the spiking neuromorphic system. Fig. 5(a)

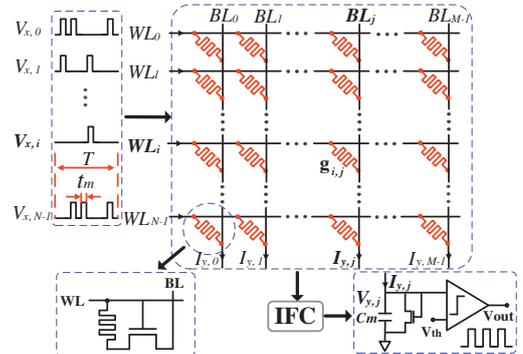


Fig. 4. The spiking neuromorphic design with a memristor crossbar array.

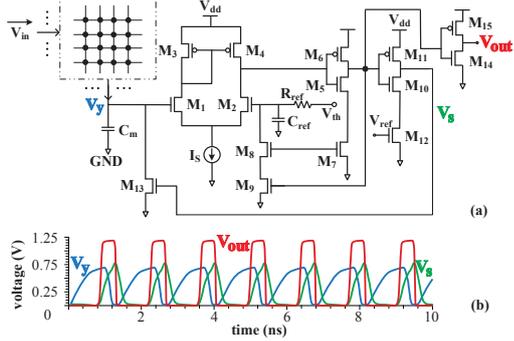


Fig. 5. The IFC circuit (a) the schematic and (b) the simulation waveforms.

depicts the schematic of a new IFC design featuring high speed and low power consumption [14]. During the operation, the BL voltage  $V_y$  continues increasing until it reaches  $V_{th}$ . Then the differential pair ( $M_1$ – $M_4$ ) together with the following two cascaded inverters ( $M_5$ – $M_7$  &  $M_{10}$ – $M_{12}$ ) generates a high voltage at  $V_s$ , which in turn enables the discharging transistor  $M_{13}$ . Consequently,  $V_y$  decreases quickly and eventually turns off  $M_{13}$ . As such, the firing of one output spike at  $V_{out}$  is completed and a new iteration of integrate-and-fire starts. To shorten the intrinsic operation delay and therefore improve the IFC throughput, a positive feedback loop ( $M_7$ – $M_9$ ) was deployed based on the traditional comparator. Another approach was to minimize the discharge time of  $C_m$  once a spike is fired out, i.e., using a large  $M_{13}$  to provide sufficient discharging current.

Fig. 5(b) shows the waveforms of  $V_y$ ,  $V_s$ , and  $V_{out}$  of the IFC design using IBM 130nm technology, under the fastest firing frequency (568.2M spikes/sec). The design area is  $175.3\mu m^2$ , which is compatible to that of traditional designs, e.g.,  $120\mu m^2$  at 65nm technology in [16]. Its energy consumption is  $0.48pJ$ -per-spike, which is about a quarter of the one in [16] ( $2pJ$ -per-spike).

The computational accuracy of the spiking neuromorphic design was evaluated based on a  $32 \times 32$  crossbar array. Here, the system computational accuracy is defined as the linearity between the obtained output spike number  $n_{y,j}$  and actual computation on the crossbar  $\sum_{i=0}^{N-1} g_{ij}\delta_i$ . Assume that an input spike has a  $2ns$  period with 50% utilization rate (that is,  $t_m = 1ns$ ). The resistance values and the input pulse numbers are randomly assigned to cover the entire input range.  $T$  varies from  $10ns$  to  $80ns$  at a step of  $10ns$  to examine the temporal scalability of the design.

It can be seen from Fig. 6 that as  $\sum_{i=0}^{N-1} g_{ij}\delta_i$  increases, the rising rate of  $n_{y,j}$  becomes smaller. This is because a larger  $\sum_{i=0}^{N-1} g_{ij}\delta_i$  and therefore a bigger  $I_{y,j}$ . It results in a faster switching of  $V_{y,j}$  from  $0V$  to  $V_{th}$ , making the impact of the IFC delay overhead  $t_0$  more prominent. Nonetheless, a good computational accuracy (i.e., output linearity) is obtained when  $\sum_{i=0}^{N-1} g_{ij}\delta_i$  is small (i.e.,  $< 0.15mS$ ). In fact, our investigation at application level also show that most of the operations of neural network implementations fall into this small range [14]. Furthermore, for different combinations of inputs and resistive array patterns with the same  $\sum_{i=0}^{N-1} g_{ij}\delta_i$ , the generated pulse number may be slightly different (no more than  $\pm 1$ ). Such a fluctuation comes from the difference in  $I_{y,j}$ 's waveform and amplitude generated by these combina-

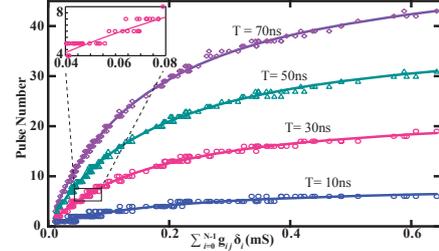


Fig. 6.  $n_{y,j}$  vs.  $\sum_{i=0}^{N-1} g_{ij}\delta_i$  under various  $T$ .

tions.

Such a spiking neuromorphic system is designed mainly for learning and classification applications whose algorithms naturally tolerate the low resolution and variability in the computations. Moreover, the imperfect output linearity shown in Fig. 6 can also be compensated during circuit implementation as long as the output spike and the weighted input spikes have a monolithic mapping relation.

#### IV. ARCHITECTURE

Recently, a highly-efficient reconfigurable neuromorphic computing accelerator with on-chip memristor-based crossbar (MBC) arrays as perceptron network, named as *RENO*, was proposed aiming at the acceleration of ANNs computations [17]. Unlike the spike-based computations where the data is represented by the pulse signals with different frequencies and amplitudes [18], the design adopts a hybrid method in data representation: the computation within MBCs and the signal communications among MBCs are conducted in analog form, while the control information remains as digital signals.

Fig. 7 illustrates the *RENO* architecture. It works as a complimentary functional unit to CPU and particularly accelerates ANN-relevant executions. In the design, *memristor-based crossbar* (MBC) arrays are used to perform high efficient analog neuromorphic computing. And a *mixed-signal interconnection network* (M-net) is developed to connect the MBCs and conduct the topological reconfiguration of *RENO*. To receive command/data and send back result in digital form to processor, *input*, *output* and *configuration FIFOs* are located at the interface of *RENO*.

MBC arrays are arranged in a *centralized mesh* (CMesh) manner to minimize the cost of the interconnection network [19]. The example in the figure includes four array groups, each of which is formed with four MBC arrays connected through a group router. A MBC array is partitioned into four sub-crossbars to implement the multiplication of the combination of the signed signals and the signed synaptic weights.

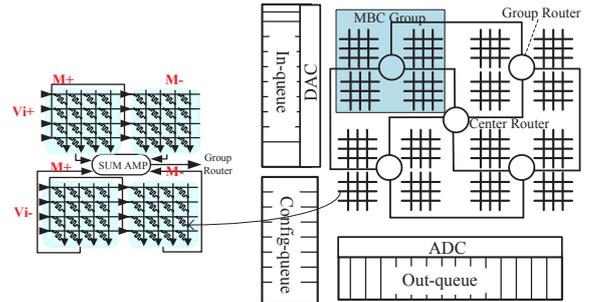


Fig. 7. The *RENO* architecture.

The optimal MBC design contains 64 rows and 64 columns which offers a good compromise between performance and reliability. Moreover, this array scale covers the majority of learning applications, 80% of which have less than 60 neurons in the input layer [20]. Applications requiring larger connection matrices can be partitioned into smaller tasks and executed on multiple MBC arrays simultaneously or sequentially.

In this centralized hierarchical architecture, the data communication is performed at both inter-group and intra-group levels. The *central router* shown in Fig. 7 connects to CPU and all *group routers*. Each group router talks to the four local MBC arrays within the group, three other group routers, and the central router. Such a centralized scheme maximizes the number of parties that each router communicates with, minimizes the effective communication distance and the hop count, mitigates the bottleneck effect of the central router, and simplifies the control complexity.

The signal transmission within RENO can be realized in either digital or analog form. Digital signal transfer has good controllability and supports high-frequency operations. However, as the computation of MBC arrays is in analog form, *digital-to-analog/analog-to-digital* (DA/AD) conversions are required at the interface of MBC arrays and routers, which inevitably degrades the signal precision and results in significant area and power overheads. The small footprint of the MBC arrays limits the data communication distance, e.g., within  $0.53mm$ , making it possible to transfer signals in analog form. Moreover, the impact of signal distortion generated during the analog signal transmission on computation reliability can be tolerated by the intrinsic high fault resistance of ANN algorithms. Instead, a mixed-signal interconnection network called *M-Net* is used to assist the task mapping and data migration in the MBC arrays. M-Net maintains the data in analog form while transfers the control and routing information in digital form so as to simplify the synchronization and communication between CPU and RENO.

A frontend scheme can be used to assist preparing RENO for ANN computation [21]. As illustrated in Fig. 8, the system frontend is composed of all the preparation steps. The codes that are already or can be implemented with ANN are identified first. Note that here the Boolean function XOR is used just for illustration purpose and the realistic target codes can be much more sophisticated. Based on the characteristics and complexity of the target codes, the topology of ANN, including the number of layers and the number of neurons at each layer etc., is decided and the ANN is trained offline. After that, the trained ANN is mapped to the NCA structure through the reconfiguration logic. During the NCA-aware compilation, the target codes are modified with the annotations of NCA IO instruction generation; then the compiler takes the topology of the trained ANN as the input parameters to generate NCA configuration instructions.

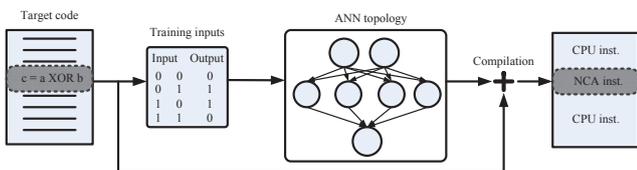


Fig. 8. The frontend for data preparation.

## V. DESIGN AUTOMATION FRAME

Although memristor crossbar is believed to be a game changing technology for neuromorphic system realization, how to efficiently design such a system with minimized (or even practical) hardware cost is still a research topic barely touched.

In application layer, for example, large neural networks are usually very sparse. In LDPC coding based on message passing algorithm, for example, the network sparsity is higher than 99% [22]. Here the sparsity of a network is defined as one minus the ratio between the number of actual connections and all possible connections in the network. In fact, such a high sparsity is also close to the biological facts that in neocortex, neurons are typically connected to only  $10^{-9}$  to  $10^{-7}$  of all the neurons and these connections are limited in the neighborhood of  $1cm^2$  of the tissue [9].

However, when the sparsity of a network is high, using memristor crossbars to implement such a network becomes inefficient, as the utilization rate of the connections in the crossbar will be low. It may be more efficient to realize these sparse connections using smaller-size crossbars or even discrete synapses. The tradeoffs between the selection of the crossbars with different sizes, the crossbar utilization rates and the impacts on physical design cost inspire this work.

An EDA flow called as AutoNCS was proposed to design a custom memristor-based neuromorphic computing systems (NCS) [23]. It is an iterative process based on spectral clustering algorithm to consolidate synapse connections into clusters and map them to memristor crossbars for high utilization rate of the connections in the crossbars. Note that implemented design can still perform various tasks because the function of a NCS can be trained by tuning the weights of the connections.

Fig. 9 depicts the overview of the design automation frame for large-scale neuromorphic computing system. It consists of the following four components:

- 1) *Modified spectral clustering* (MSC) that groups the connections in a network into dense clusters that can be efficiently mapped to memristor crossbars;
- 2) *Greedy cluster size prediction* (GCP) that constrains the largest cluster size within the maximum available crossbar scale;
- 3) *Iterative spectral clustering* (ISC) that repeatedly performs clustering on the networks to group the connections into clusters, and minimize the outliers that need to be mapped to discrete synapses; and
- 4) a customized physical design method to realize the neuromorphic systems based on the clustering result.

A testbench of sparse Hopfield network was used to evaluate AutoNCS. The network with a size of 500 was trained for

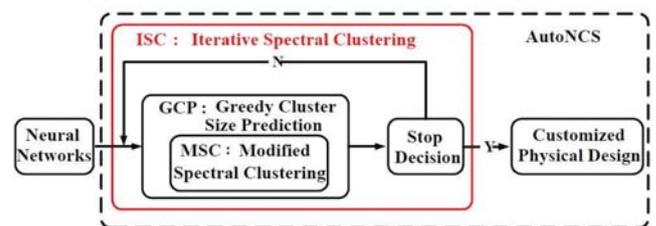


Fig. 9. The overview of the EDA frame.

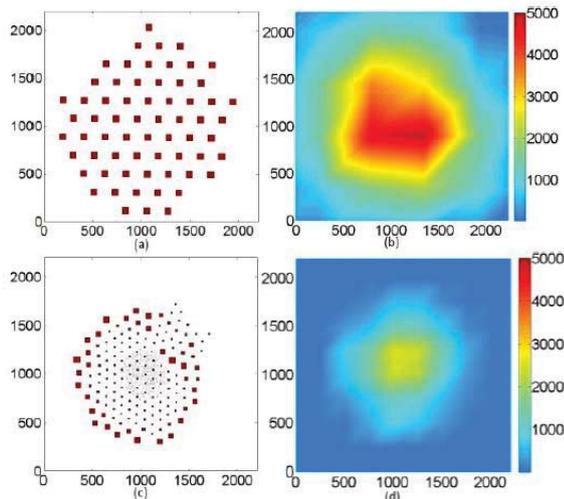


Fig. 10. The placement and routing results of the Hopfield testbench without clustering are shown in (a) and (b). The results with AutoNCS are shown in (c) and (d).

recognition of 30 patterns. While offering a recognition rate above 90%, the sparsity of the network is 94.39%. Fig. 10 compares the optimal placement and routing results in full crossbar (FullCro) and AutoNCS. In optimal FullCro, crossbars with the maximum size are uniformly placed, resulting in heavy wire congestion in the center. However, in AutoNCS, large crossbars on the periphery realized the majority of connections, leaving only sparse connections implemented by small crossbars and discrete synapses in the inner place. This topology reduces wirelength, area and average delay substantially.

## VI. CONCLUSION

The emerging memristor technology has demonstrated great potential in neuromorphic system design for its similar behavior to biological synapse, nonvolatile data storage, reconfigurability, analogy process capability, as well as the extreme high connectivity. This paper gives a brief summary on the research activities in utilizing the memristor crossbar arrays for neuromorphic design. Holistic effects across different areas shall be integrated and there are still many problems to be solved to obtain a practical neuromorphic hardware for large-scale applications.

## ACKNOWLEDGMENT AND DISCLAIMER

This work is supported in part by NSF 1337198 and DARPA D13AP00042. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF, DARPA, or their contractors.

## REFERENCES

- [1] J. Partzsch and R. Schuffny, "Analyzing the scaling of connectivity in neuromorphic hardware and in models of neural networks," *IEEE Transactions on Neural Networks (TNNLS)*, vol. 22, no. 6, pp. 919–935, June 2011.
- [2] S. Li, C. Wu, H. Li, B. Li, Y. Wang, and Q. Qiu, "Fpga acceleration of recurrent neural network based language model," in *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, May 2015, pp. 111–118.

- [3] B. Li, E. Zhou, B. Huang, J. Duan, Y. Wang, N. Xu, J. Zhang, and H. Yang, "Large scale recurrent neural network on gpu," in *2014 International Joint Conference on Neural Networks (IJCNN)*, July 2014, pp. 4062–4069.
- [4] M. Hu, H. Li, Q. Wu, and G. Rose, "Hardware realization of bsb recall function using memristor crossbar arrays," in *49th ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2012, pp. 498–503.
- [5] M. Hu, H. Li, Y. Chen, Q. Wu, G. Rose, and R. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1864–1878, Oct 2014.
- [6] L. Chua, "Memristor—the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sep 1971.
- [7] R. Williams, "How we found the missing memristor," *IEEE Spectrum*, vol. 45, no. 12, pp. 28–35, Dec 2008.
- [8] X. Wang, Y. Chen, H. Xi, H. Li, and D. Dimitrov, "Spintronic Memristor Through Spin-torque-induced Magnetization Motion," *IEEE Electron Device Letters*, vol. 30, pp. 294–297, 2009.
- [9] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Letters*, vol. 10, no. 4, pp. 1297–1301, 2010.
- [10] D. Kim, S. Seo, S. Ahn, D. Suh, M. Lee, B. Park, I. Yoo, I. Baek, H. Kim, E. Yim *et al.*, "Electrical observations of filamentary conduction for the resistive memory switching in NiO films," *Applied physics letters*, vol. 88, no. 20, pp. 202 102–202 102, 2006.
- [11] M. Hu, H. Li, Y. Chen, X. Wang, and R. Pino, "Geometry variations analysis of TiO<sub>2</sub> thin-film and spintronic memristors," in *Proceedings of the 16th Asia and South Pacific Design Automation Conference*, 2011, pp. 25–30.
- [12] W. Yi, F. Perner *et al.*, "Feedback Write Scheme for Memristive Switching Devices," *Applied Physics A*, vol. 102, no. 4, pp. 973–982, 2011.
- [13] G. Medeiros-Ribeiro, F. Perner, R. Carter, H. Abdalla, M. D. Pickett, and R. S. Williams, "Lognormal Switching Times for Titanium Dioxide Bipolar Memristors: Origin and Resolution," *Nanotechnology*, vol. 22, no. 9, p. 095702, 2011.
- [14] C. Liu, B. Yan, C. Yang, L. Song, Z. Li, B. Liu, Y. Chen, H. Li, Q. Wu, and H. Jiang, "A spiking neuromorphic design with resistive crossbar," in *Proceedings of the 52nd Annual Design Automation Conference*, 2015, pp. 14.1–14.6.
- [15] W. Gerstner and W. M. Kistler, *Spiking Neuron Models*. Cambridge University Press, 2002.
- [16] A. Joubert *et al.*, "Hardware spiking neurons design: Analog or digital?" in *IJCNN*, June 2012, pp. 1–5.
- [17] X. Liu, M. Mao, B. Liu, H. Li, Y. Chen, B. Li, Y. Wang, H. Jiang, M. Barnell, Q. Wu, and J. Yang, "Reno: A high-efficient reconfigurable neuromorphic computing accelerator design," in *Proceedings of the 52nd Annual Design Automation Conference*, 2015, pp. 66.1–66.6.
- [18] A. S. Cassidy, P. Merolla, J. V. Arthur, S. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman, A. Amir, D. B. dayan Rubin, E. Mcquinn, W. P. Risk, and D. S. Modha, "Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores," in *IJCNN*, 2013.
- [19] J. Balfour and W. J. Dally, "Design tradeoffs for tiled cmp onchip networks," in *ICS*, 2006, pp. 187 – 198.
- [20] O. Temam, "A defect-tolerant accelerator for emerging high-performance applications," in *ISCA*, 2012, pp. 356–367.
- [21] X. Liu, M. Mao, H. Li, Y. Chen, H. Jiang, J. Yang, Q. Wu, and M. Barnell, "A heterogeneous computing system with memristor-based neuromorphic accelerators," in *2014 IEEE High Performance Extreme Computing Conference (HPEC)*, 2014, pp. 1–6.
- [22] "Ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 10: Mesh networking," *IEEE Std 802.11s-2011*, pp. 1–372, Sept 2011.
- [23] W. Wen, C.-R. Wu, X. Hu, B. Liu, T.-Y. Ho, X. Li, and Y. Chen, "An eda framework for large scale hybrid neuromorphic computing systems," in *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.