Bio-Inspired Computing with Resistive Memories – Models, Architectures and Applications

Qing Wu
Information Directorate
Air Force Research Laboratory
Rome, NY, USA
qing.wu.2@us.af.mil

Beiye Liu, Yiran Chen, Hai Li
Elec. and Comp. Engineering
University of Pittsburgh
Pittsburgh, PA, USA
{bel34, yic52, hal66}@pitt.edu

Qiuwen Chen, Qinru Qiu
Elec. Engineering & Comp. Science
Syracuse University
Syracuse, NY, USA
{qchen14, qiqiu}@syr.edu

***Abstract*—The traditional Von Neumann architecture has constrained the potential for applying massively parallel architecture to embedded high performance computing where we must optimize the size, weight and power of the system. Inspired by highly parallel biological systems, such as the human brain, the neuromorphic architecture offers a promising novel computing paradigm for compact and energy efficient platforms. The discovery of memristor devices provided the element we need with unprecedented efficiency in realizing such a computing architecture. There are still many challenges left to meet our goal of a fully functional bio-inspired computer. Here we will discuss our research in memristor crossbar based architecture, adaptation of this architecture for cogent confabulation models, and potential applications of the bio-inspired computer.**

Keywords—neuromorphic; architecture; memristor; bio-inspired; confabulation

I. INTRODUCTION

The explosion of “big data” applications imposes severe challenges of processing speed and scalability on traditional computers. The performance of the Von Neumann machine is hindered by the increasing performance gap between CPU and memory (known as the “memory wall”), motivating the active research on new or alternative computing architecture. As one important example, bio-inspired (or neuromorphic) computing systems have gained considerable attention.

Bio-inspired computing systems refer to the computing architectures inspired by the working mechanism of the human brain. The human neocortex system naturally possesses a massively parallel architecture with closely coupled memory and computing as well as unique analog domain operations [1]. The simple unified building blocks (i.e., neurons) follow integrate-and-fire mechanisms, leading to an ultra-high computing performance beyond 100 TOPS (Trillion Operations Per Second) and a power consumption of a mere 20 Watts. By imitating such structures, neuromorphic computing systems are anticipated to be superior to the conventional computer systems in tasks such as image recognition and natural language understanding. As the most resource-consuming part in neuromorphic algorithms [2], matrix operations are normally processed by hardware accelerators like CPU/GPU/FPGA [3] or VLSI circuits [4]. The straightforward hardware realization of neural networks, however, commonly consumes a large volume of memory and computing resources, incurring high design complexity and hardware cost.

The structural similarity makes the reconfigurable array conceptually efficient for matrix operations [16][17][18] and inspired many researches on the corresponding circuit designs, i.e., the arrays of flash transistor [20] or DRAM capacitor [21]. However, the computation capacity and scalability of these designs are generally limited by the large cell footprint. Recently, the discovery of the memristor device triggered a revolution in neuromorphic computing system design: synaptic behavior is easily mimicked by the historical recording property of the memristor while the crossbar array structure offers the highest integration density in 2D/3D designs [22].

In this paper, we propose a novel information processing system that combines the flexibility of conventional architecture in logic-scientific computation with the efficiency of neuromorphic architecture for applications in the domains of language understanding and data analytics. By leveraging associative memory and inference-based information processing models, a bio-inspired computing architecture is introduced to accelerate neuromorphic computations with ultra-low energy consumption. The proposed bio-inspired computing method integrates a wide spectrum of new technologies in device, circuit, systems, models and applications. Using the memristor devices [5][6] as the building blocks, we propose a memristor crossbar-based analog circuit design as the basic computing component for neuromorphic models such as associative memory and inference. At the system level we propose a heterogeneous computing architecture and memory hierarchy across digital and analog (neuromorphic) domains that facilitate seamless coordination between conventional pipeline and neuromorphic accelerators. We will also introduce a large scale application that can benefit significantly from the proposed hardware architecture.

The remainder of the paper is organized as follows. Section II introduces the operating principles of the memristor device. Section III describes a circuit design based on the memristor crossbar arrays that realizes an approximation of matrix-vector multiplication computing. Section IV introduces the proposed heterogeneous system architecture. An application in autonomous large area traffic monitoring will be discussed in Section V. Section VI provides conclusions of the work.

II. THE MEMRISTOR DEVICE

The existence of the memristor was predicted in circuit theory about forty year ago [5]. In 2008, the physical realization of a memristor was firstly demonstrated by HP Lab through a TiO_2 thin-film structure [6]. Afterwards, many memristive materials and devices have been rediscovered. Intrinsically, a memristor behaves similarly to a synapse: it can “remember” the total electric charge/flux ever to flow through it [8][9]. Moreover, memristor-based memories can achieve a very high integration density of 100 Gbits/cm^2, a few times higher than flash memory technologies [7]. These unique properties make it a promising device for massively-parallel, large-scale neuromorphic systems [10][11].

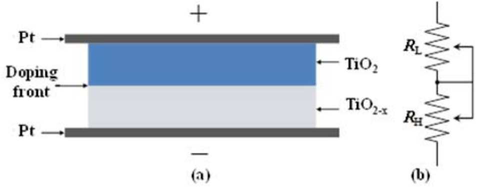

Fig. 1. TiO_2 memristor: (a) structure; (b) equivalent circuit.

Fig. 1 illustrates the conceptual view of the TiO_2 thin-film memristor and the corresponding variable resistor model, which is equivalent to two serially-connected resistors. Here, R_L and R_H respectively denote the *low resistance state* (LRS) and the *high resistance state* (HRS). The overall memristance can be expressed as:

$$M(p) = p \cdot R_H + (1-p) \cdot R_L \tag{1}$$

where p ($0 \leq p \leq 1$) is the relative doping front position, which is the ratio of doping front position over the total thickness of the TiO_2 thin-film. The velocity of doping front movement $v(t)$, driven by the voltage applied across the memristor $V(t)$, can be expressed as:

$$v(t) = \frac{dp(t)}{dt} = \mu_v \cdot \frac{R_L}{h^2} \cdot \frac{V(t)}{M(p)} \tag{2}$$

where μ_v is the equivalent mobility of dopants, h is the total thickness of the thin film, and $M(p)$ is the total memristance when the relative doping front position is p. In general, a certain energy (or threshold voltage) is required to enable the state change in a memristive device [12]. When the electrical excitation through a memristor is greater than the threshold voltage, i.e., $V(t) > V_{th}$, the memristance changes (in training). Otherwise, a memristor behaves like a resistor.

III. Computing With Memristor Crossbar CircuitS

Crossbar array illustrated in Fig. 2 is a typical structure of memristor-based memories. It employs a memristor device at each intersection of horizontal and vertical metal wires without any selectors [15]. The memristor crossbar array is naturally attractive for implementation of connection matrix in neural networks for it can provide a large number of signal connections within a small footprint and conduct the weighted combination of input signals [13][14].

As shown in Fig. 2, the N-by-M memristor crossbar array is a basic building block to achieve matrix-vector multiplication approximation computation functionality. A set of input voltages $\mathrm{VI}^{\mathrm{T}} = \{\mathrm{vi}_1, \mathrm{vi}_2, \cdots, \mathrm{vi}_{\mathrm{N}}\}$ are applied on each of the N *word-lines* (*WL*s) of the array, and the current is collected through each of the M bit-lines (*BL*s) by measuring the voltage across a sensing resistor. The same sensing resistors are used on all the BLs with resistance r_s, or conductance $g_s = 1/r_s$. The output voltage vector is: $\mathrm{VO}^{\mathrm{T}} = \{\mathrm{vo}_1, \mathrm{vo}_2, \cdots, \mathrm{vo}_{\mathrm{M}}\}$. The memristor sitting on the connection between WL_j and BL_i has a memristance of $m_{i,j}$. The corresponding conductance is $g_{i,j} = 1/m_{i,j}$. Then the relation between the input and output voltages can be approximated by:

$$\mathrm{VO} \cong \mathrm{C} \times \mathrm{VI} \tag{3}$$

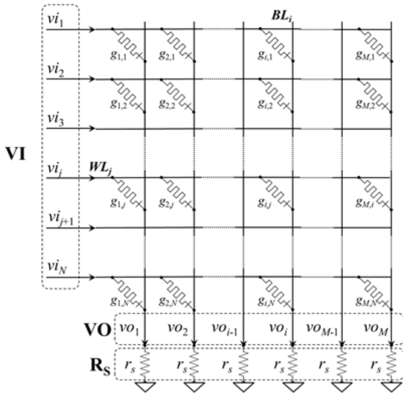

Fig. 2. A memristor crossbar array.

Here, $\mathbf{C}$ is an M-by-N matrix that can be represented by the memristors and the sensing (load) resistors as:

$$\mathbf{C} = \mathbf{D} \times \mathbf{G} = diag(d_1, \cdots, d_M) \times \begin{bmatrix} g_{1,1} & \cdots & g_{1,N} \\ g_{2,1} & & g_{2,N} \\ \vdots & \ddots & \vdots \\ g_{M,1} & \cdots & g_{M,N} \end{bmatrix} \tag{4}$$

where $\mathbf{D}$ is a diagonal matrix with diagonal elements of:

$$d_i = 1/(g_s + \sum_{k=1}^{N} g_{i,k}),\ i = 1, 2, \ldots, M. \tag{5}$$

Eq. (3) indicates that a trained memristor crossbar array can be used to construct the connection matrix $\mathbf{C}$, and transfer the input vector $\mathbf{VI}$ to the output vector $\mathbf{VO}$.

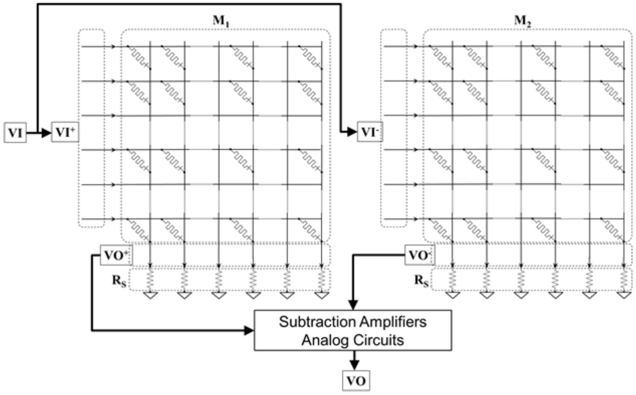

Fig. 3. Matrix-vector multiplication approximation design.

Given a general matrix $\mathbf{A}$ with both positive and negative elements, we can split the positive and negative elements of $\mathbf{A}$ into two matrixes $\mathbf{A}^+$ and $\mathbf{A}^-$ as:

$$a_{i,j}^{+} = \begin{cases} a_{i,j}, & if\ a_{i,j} > 0 \\ 0, & if\ a_{i,j} \leq 0 \end{cases} \text{ and } a_{i,j}^{-} = \begin{cases} 0, & if\ a_{i,j} > 0 \\ -a_{i,j}, & if\ a_{i,j} \leq 0 \end{cases} \tag{6}$$

As such, a general matrix-vector multiplication approximation becomes:

$$\mathbf{VO} = \mathbf{A}\cdot\mathbf{VI} = \mathbf{A}^{+}\cdot\mathbf{VI} - \mathbf{A}^{-}\cdot\mathbf{VI} \qquad (7)$$

Here, the two matrices $\mathbf{A}^{+}$ and $\mathbf{A}^{-}$ can be mapped to two memristor crossbar arrays $\mathbf{M}_1$ and $\mathbf{M}_2$ in a scaled version $\widehat{\mathbf{A}}^{+}$ and $\widehat{\mathbf{A}}^{-}$, respectively. Fig. 3 shows the circuit diagram. This memristor crossbar-based design has been used to realize associative memory model training and recall operations. Details of the implementations, algorithms and results can be found in [16][17][18].

IV. Heterogeneous Digital-Neuromorphic System

We named the dual memristor crossbar based design the *Neuromorphic Computing Accelerator* (NCA). The NCA is much faster and more energy efficient than conventional computing architectures such as CPU (10~20 GFLOPS/W) and GPGPU (20~30 GFLOPS/W). The major cause of the operational latency of an NCA comes from the interconnects of crossbar arrays and peripheral circuits. For instance, to complete one iteration in an associative memory recall operation for a 256-entry vector, the computation latency of an NCA should not exceed 100ns even if we conservatively assume the delay of a summing amplifier is 50ns (including setup and computation). As a comparison, the delay of the Boolean mapping is 750ns by assuming using 256 4-bit multipliers with 2ns latency and 256 4-bit CLA adders with 1ns latency. Fig. 4 shows the estimated power efficiency of memristor-based NCA at 65nm technology node. The NCA design with a small crossbar array, i.e., 16×16, already obtains 1,200 GFLOPS/W (billion floating point operations per second per watt). Increasing the crossbar size to 256×256 results in much higher computation parallelism, and further boosts the power efficiency close to 2,000 GFLOPS/W.

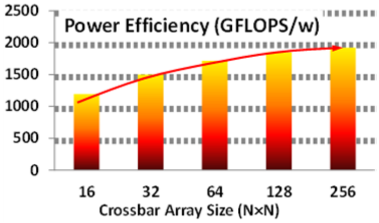

Fig. 4. Estimated power efficiency of the NCA.

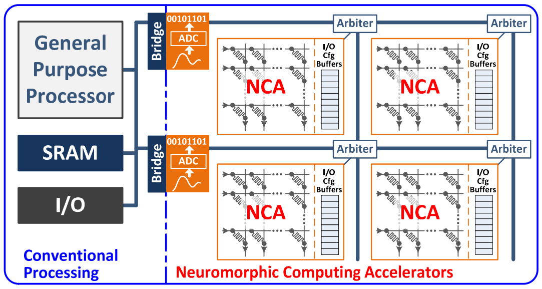

Fig. 5. Conceptual diagram of a digital-neuromorphic system.

Fig. 5 shows the conceptual design of the proposed heterogeneous system with both conventional pipeline and crossbar-based NCAs. The control signals and data communications between the NCAs are through the arbiters. The data transferring could be in either digital and/or analog forms, though the analog one offers the maximum performance. The conversion between analog and digital formats is constrained at the interface between the conventional pipeline and the NCA array.

V. An Application of Neuromorphic MethodS for Autonomous Large Area Traffic Monitoring

In this section, we introduce recent work applying neuromorphic computing models and algorithms in an autonomous anomaly recognition and detection (AnRAD) framework. The proposed framework is based on cogent confabulation [19], which is a computation model that mimics human information processing. The model has successfully been applied in sentence completion and document image recognition [23]. In this framework, the large area is first partitioned into smaller zones (as shown in Fig. 6) that can be independently processed. Then, a *knowledge base* (KB) is built for each zone by feeding traffic records into properly modeled knowledge networks. When new traffic information is received, anomaly scores will be calculated by means of likelihood-ratio test for the observed events. Events with high anomaly scores will be marked as potential anomalies and alarms will be sent to the human observer.

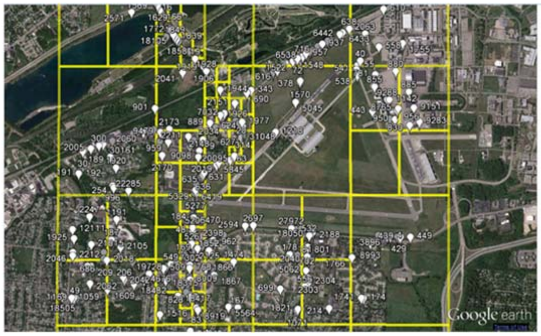

Fig. 6. Operational overview of the AnRAD framework.

The confabulation model represents the observation using a set of features. These features construct the basic dimensions that describe the world of applications, e.g. vehicle speed and coordinates. Different observed attributes of a feature are referred to as *symbols*. The set of symbols used to describe the same feature forms a *lexicon* and the symbols in a lexicon are exclusive to each other. *Knowledge links* (*KL*) are established among lexicons. They are directed edges from the source lexicons to target lexicons. Each knowledge link is associated with a matrix. The *ij*th entry of the matrix gives the conditional probability $\log\left[p\left(s_i\middle|t_j\right)\right]$ between the symbols s_i in the source lexicon and t_j in the target lexicon. The knowledge matrix is constructed during training by extracting and associating features from the inputs.

The excitation of a symbol *t* in lexicon *l* is calculated by summing up all incoming knowledge links:

$$el(t) = \sum_{k\in F_l}\left(\sum_{s\in S_k} I(s)\ln\left(\frac{p(s|t)}{p_0}\right) + B\right) \qquad (1)$$

where F_l denotes the set of lexicons that have connections to *l*, and S_k is a set that consists the collections of symbols in lexicon *k*; *I(s)* is the firing strength of source symbol *s*, and it is set to 1 if *s* is observed without ambiguity; p_0 is the minimum probability that is considered informative. Parameter *B* is a constant called *band gap*, it is 0 if none of the active source symbols in S_k has knowledge links going into t. The band gap ensures that symbols with more KLs receive higher excitation over those with fewer KLs.

Fig. 7 shows the task graph and data flow for implementing cogent confabulation on the proposed heterogeneous hardware platform. The computation model consists of two types of operations; matrix-vector multiplication (MVM) and integrate-and-firing (IF) operation. Let vectors R and E denote the input and output of MVM with dimensions M and N respectively, then $E = \boldsymbol{KL}*R$, where $\boldsymbol{KL}$ is an $M\times N$ matrix. Each feature extraction engine associates with an MVM. Each knowledge link also accompanies an MVM. Its input is the vector of source lexicon's firing strength and its output is the excitation coming from this knowledge link. Each IF operator corresponds to a lexicon in the inference layer. It has N inputs E_i, $1\leq i\leq N$, and one output R, all of which are N dimensional vectors, where N is the number of neurons in that lexicon. Each input is a vector of excitations from a knowledge link. The output vector R gives the firing strength of neurons in the lexicon. The IF operator first calculates the excitation level of lexicon by adding all input vectors, $E = \sum_{i=1}^{N} E_i$. Then it suppresses the least excited neurons by clearing the smallest entries in E. Finally it normalizes all non-zero entries in E and outputs the result.

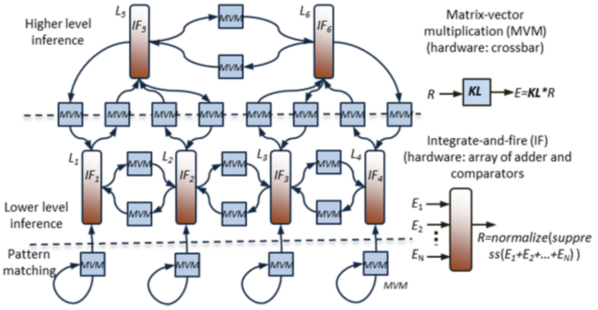

Fig. 7. Task graph and data flow of implementing the cogent confabulation model on the heterogeneous system.

VI. Conclusions

We have proposed a bio-inspired computing architecture that leverages associative memory and inference-based information processing models to accelerate neuromorphic computations with ultra-low energy consumption. Using the memristor devices as the building blocks, we proposed a memristor crossbar-based analog circuit design as the basic computing component for neuromorphic models such as associative memory and inference. At the system level, a heterogeneous computing architecture and memory hierarchy is introduced for facilitating seamless coordination between conventional pipeline and neuromorphic accelerators.

References

[1] H. Moravec, "When will Computer Hardware Match the Human Brain?" *Journal of Evolution and Technology*, Vol. 1, no. 1, pp. 10, 1998.

[2] M. Wang, B. Yan, J. Hu, and P. Li, "Simulation of Large Neuronal Networks with Biophysically Accurate Models on Graphics Processors," The 2011 International Joint Conference on Neural Networks (IJCNN), pp. 3184–3193, 2011.

[3] H. Shayani, P. Bentley, and A. Tyrrell, "Hardware Implementation of A Bio-plausible Neuron Model for Evolution and Growth of Spiking Neural Networks on FPGA," NASA/ESA Conference on Adaptive Hardware and Systems, pp. 236–243, 2008.

[4] K. Maezawa, T. Akeyoshi, and T. Mizutani, "Functions and Applications of Monostable-Bistable Transition Logic Elements (Mobile) Having Multiple-Input Terminals," IEEE Electron Device Letters, Vol. 41, pp. 148–154, 1994.

[5] L. Chua, "Memristor-the missing circuit element," *IEEE Transaction on Circuit Theory*, vol. 18, 1971, pp. 507–519.

[6] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[7] Y. Ho, G.M. Huang, and P. Li, "Nonvolatile memristor memory: device characteristics and design implications," in *International Conference on Computer-Aided Design (ICCAD)*, 2009, pp.485–490.

[8] M. Di Ventra, Y.V. Pershin and L.O. Chua, "Circuit elements with memory: memristors, memcapacitors, and meminductors," *Proceedings of the IEEE*, vol. 97, no. 10, pp. 1717–1724, 2009.

[9] L. Chua, "Resistance switching memories are memristors," *Applied Physics A: Materials Science& Processing*, vol. 102, no. 4, pp. 765–783, 2011.

[10] Q. Xia, W. Robinett, M. W. Cumbie, N. Banerjee, T. J. Cardinali, J. J. Yang, W. Wu, X. Li, W. M. Tong, D. B. Strukov, G. S. Snider, G. Medeiros-Ribeiro, and R. S. Williams, "Memristor-CMOS hybrid integrated circuits for reconfigurable logic," *Nano letters*, vol. 9, no. 10, pp. 3640–3645, 2009.

[11] 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.

[12] Y. Pershin and M. Di Ventra, "Practical approach to programmable analog circuits with memristors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 8, pp. 1857–1864, 2010.

[13] U. Ramacher and C. V. D. Malsburg, *On the Construction of Artificial Brains*. Springer, 2010.

[14] T. Hasegawa, T. Ohno, K. Terabe, T. Tsuruoka, T. Nakayama, J. K. Gimzewski, and M. Aono, "Learning abilites achieved by a single solid-state atomic switch," *Advanced Materials*, vol. 22, no. 16, pp. 1831–1834, 2010.

[15] A. Heittmann and T. G. Noll, "Limits of writing multivalued resistances in passive nano-electronic crossbars used in neuromorphic circuits," *ACM Great Lakes Symposium on VLSI (GLSVLSI)*, 2012, pp. 227–232.

[16] M. Hu, H. Li, Q. Wu, and G. Rose, "Hardware Realization of Neuromorphic BSB model with memristor crossbar network," IEEE Design Automation Conference (DAC), pp. 554–559, 2012.

[17] M. Hu, H. Li, Q. Wu, G. Rose, and Y. Chen, "Memristor Crossbar Based Hardware Realization of BSB Recall Function," International Joint Conference on Neural Networks (IJCNN), pp. 1-7, 2012.

[18] M. Hu, H. Li, Y. Chen, G. Rose, and Q. Wu, "BSB Training Scheme Implementation on Memristor-Based Circuit," 2013 Symposium Series on Computational Intelligence, April 2013.

[19] R. Hecht-Nielsen, "Confabulation Theory: The Mechanism of Thought," Springer, August 2007.

[20] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nW/MHz, 128 Times;32 Currentmode Programmable Analog Vector-matrix Multiplier with Over Two Decades of Linearity," Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004, pp. 651 – 654, Oct. 2004.

[21] R. Genov, S. Chakrabartty, and G. Cauwenberghs, "Silicon Support Vector Machine with OnLine Learning," International Journal of Pattern Recognition and Artificial Intelligence, Vol. 17, No. 3, 385-404, 2003.

[22] C.H. Cheng, "Novel Ultra-low power RRAM with good endurance and retention," 2012 Symposium on VLSI Technology (VLSIT), June 2010, pp. 85-86.

[23] Q. Qiu, Q. Wu, M. Bishop, R. Pino, and R. W. Linderman, "A Parallel Neuromorphic Text Recognition System and Its Implementation on a Heterogeneous High Performance Computing Cluster," *IEEE Transactions on Computers*, Feb. 2012.