A Memristor Crossbar Based Computing Engine Optimized for High Speed and Accuracy

Chenchen Liu, Qing Yang, Bonan Yan, Xiaocong Du, Hai (Helen) Li
Department of Electrical and Computer Engineering, University of Pittsburgh
{chl192, qiy21, boy21, xid22, hal66}@pitt.edu
Jianlei Yang[†], Weijie Zhu[‡], Hao Jiang[‡], Qing Wu[§], Mark Barnell[§]
[†]Beihang University, [‡]San Francisco State University, [§]US Air Force Research Laboratory
[†]jerryyang@gmail.com, [‡]wzhu@mail.sfsu.edu, [‡]jianghao@sfsu.edu, [§]{qing.wu.2, mark.barnell.1}@us.af.mil

***Abstract*—Matrix-vector multiplication, as a key computing operation, has been largely adopted in applications and hence greatly affects the execution efficiency. A common technique to enhance the performance of matrix-vector multiplication is increasing execution parallelism, which results in higher design cost. In recent years, new devices and structures have been widely investigated as alternative solutions. Among them, memristor crossbar demonstrates a great potential for its intrinsic support of matrix-vector multiplication, high integration density, and built-in parallel execution. However, the computation accuracy and speed of such designs are limited and constrained by the features of crossbar array and peripheral circuitry. In this work, we propose a new memristor crossbar based computing engine design by leveraging a current sensing scheme. High operation parallelism and therefore fast computation can be achieved by simultaneously supplying analog voltages into a memristor crossbar and directly detecting weighted currents through current amplifiers. The performance and effectiveness of the proposed design were examined through the implementation of a neural network for pattern recognition based on MNIST database. Compared to a prior reported design, ours increases the recognition accuracy 8.1% (to 94.6%).**

***Keywords*—memristor crossbar, current sensing, matrix-vector computation.**

I. Introduction

The matrix-vector multiplication has been widely used in many scientific computing and engineering applications. Examples include the linear system solvers in economic modeling and control system simulation [1]. It is also the most critical component of machine learning and deep neural network models [2]. A common approach for the performance enhancement of matrix-vector multiplications is to increase the execution parallelism, which requires a large number of computation resources and leads to high energy consumption [3][4]. Moreover, in nowadays big data environment, data generation and collection grow in an exponential rate, which essentially boosts up the scales of the aforementioned modelings and algorithms. Thus, finding a new solution of matrix computation with better computation and energy efficiency becomes crucial.

In addition to continuous efforts on circuit and architecture development in conventional CMOS domain [5], the use of new devices has been extensively explored [6]. For instance, the discovery of memristor devices brought a great opportunity of new computing engines. Thirty-seven years after Professor Chua's theoretical prediction [7], The existence of memristor device was firstly reported by HP Labs in 2008 [8]. Memristor features non-volatility, high integration density, and multi-level (or continuous state) storage, making it a promising device for data storage [9][10] and computing [11].

As illustrated in Fig. 1, a high similarity exists between a mathematical matrix and a memristor crossbar array. Thus, the matrix-vector multiplication can be naturally realized through a high-density memristor crossbar array, by representing each matrix element with the conductance of the corresponding memristor cell. Input signals can be supplied in parallel to wordlines of the array as an input vector, and the outputs can be collected at bitlines simultaneously. Such a design concept has been investigated and demonstrated in neuromorphic systems and approximating computation [12][13][14], offering a new design scenario with high computation efficiency. Lately, a neuromorphic system with a transistor-free metal-oxide memristor crossbar performing 30 synaptic weights was demonstrated [15].

The accuracy and power consumption of memristor crossbar based matrix-vector multiplication are greatly related to the architecture and circuit selection. For example, Hu *et al.* adopted a voltage-based sensing scheme which uses voltage amplitude to represent input and output data [12]. This design provides high computation speed but the *analog-digital* and *digital-analog* (AD/DA) signal conversion introduce high signal distortion and power consumption. In spiking neuromorphic system implementation [16], the input information is supplied in the spiking format. The matrix computation result is detected by an *integrate-and-fire circuit* (IFC) and represented by output spikes. Such an approach demonstrates very high power efficiency. However, the output of memristor crossbar will not follow a linear function with the input data, due to the charge accumulation and release of IFC. Computing accuracy and speed are constrained by the limited spike number. Moreover, to better control the memristor conductance and promise computation accuracy, it is preferable to integrate a transistor with each memristor device [17]. As such, the unique $4F^2$ unite cell size (where, F is the technology feature size) cannot be maintained. And a shift in the target memristor conductance is inevitable, which increase the difficulty and complexity in matrix mapping [16][18].

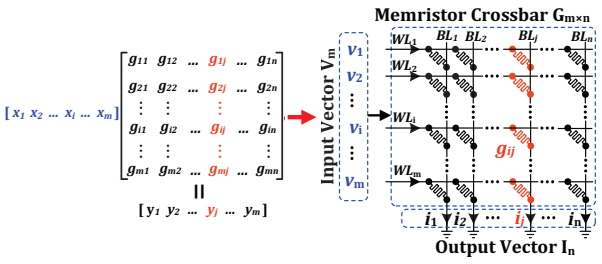

Fig. 1. Mapping a matrix-vector multiplication to a memristor crossbar.

978-1-4673-9039-2/16 $31.00 © 2016 IEEE
DOI 10.1109/ISVLSI.2016.46

In this work, we propose a new memristor crossbar based computing engine which leverages a current-based sensing scheme for higher computation speed and accuracy. The design supplies analog voltage signals to wordlines in parallel. A current buffer amplifier directly senses out bitline current so the tail voltage of bitline maintains at a constant level. We developed the matrix-vector computation engine based on a 144×144 memristor-based crossbar and current buffer amplifier at 130nm technology node. The impact of sneak path leakage and wire resistance of memristor crossbar were considered and analyzed. The performance evaluation of the proposed computing engine was conducted through a three-layer neural network for MNIST handwritten digit recognition [19]. Comparing with a latest reported design [16], the proposed work obtained 8.1% improvement in recognition accuracy and improve operation speed 40.1%.

II. Background

Memristor is a two terminal nonvolatile device that represents its state as a resistance value (or *memristance*). By carefully controlling the amplitude and duration of a memristor's external excitation (*e.g.*, voltage or current), its memristance can be changed gradually [17].

Memristors are usually organized in crossbar arrays with extremely high storage density. As illustrated in Fig. 1, the cross-point of every horizontal *wordline* (WL) and vertical *bitline* (BL) locates a memristor. Instinctively, such an array structure can be used to realize a matrix-vector multiplication, *e.g.*, $\overrightarrow{Y_n}^T = \overrightarrow{X_m}^T \times M_{m\times n}$. More specific, we can use a crossbar array $G_{m\times n}$ to denote $M_{m\times n}$ by making $g_{i,j}$, the conductance of the cell at the cross-point of WL_i and BL_j, represent the corresponding data in $M_{M\times N}$. The matrix-vector multiplication is then transformed to

$$\overrightarrow{I_n}^T = \overrightarrow{V_m}^T \times G_{m\times n}. \tag{1}$$

Where, the input vector $\overrightarrow{V_m}^T = [v_1, v_2, \cdots, v_i, \cdots, v_m]$ is composed of analog voltages to WLs. The output current i_j at the end of BL_j produces a dot production, such as

$$i_j = \sum_{i=1}^{M} g_{i,j} \cdot v_i. \tag{2}$$

Thus, all the BLs currents form the output vector $\overrightarrow{I_n}^T = [i_1, i_2, \cdots, i_j, \cdots, i_n]$. For the high integration density and parallel operation, memristor crossbars greatly improve the efficiency in matrix computation and therefore inspired extensive studies on the hardware implementation and applications [12][20][21].

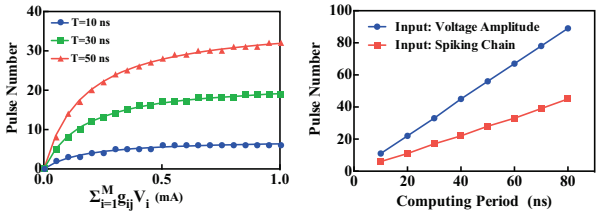

Fig. 2. (a) The computation accuracy analysis of a spiking-based design [16]. (b) The relations of output spike number vs. computing period when representing input data by spike chain or by voltage amplitude.

III. Motivation of Our Work

A memristor crossbar based computing engine senses out BL voltage/current as the computation result. Recently, Liu *et al.* implemented a spiking-based design for high power efficiency [16]. It uses an *integrate and fire circuit* (IFC) to detect the BL current and represent it by spike number. The charging and discharging overhead of IFC results in a non-linear relationship between the ideal BL current as a sum of weighted multiplication ($\sum_{i=1}^{M} g_{ij}V_i$) and the output spiking number: with BL current increase, the output spike number will increase first and then start to saturate, as shown in Fig. 2(a). It can also be observed from the figure that prolonging the computing period, *e.g.*, increasing it from 10ns to 50ns, will help produce more output spike number, resulting in better linearity and higher computation accuracy.

Moreover, the input data in the spiking-based design [16] is converted to a chain of spikes: the spike frequency (*i.e.*, the number of spikes within a constant computing period) represents the scale of data. Such a digitalized interface guarantees the good noise immunity and high energy efficiency in signal transferring. However, it also induce a low utilization rate at time domain. Thus, the computing period need to be sufficiently long to satisfy the computation accuracy requirement. In contrast, when using voltage amplitude to represent the strength of input data as [12], the computing period will be fully utilized and therefore could be a lot shorter. For example, we compared the relations of the output spike number and the computing period of two designs and Fig. 2(b) shows the results when the BL current is set to 0.6mA. To generate the same amount of output spikes, the computing period of the design representing input data by voltage amplitude is only about the half of the version using an input spike chain.

We also note that the spiking-based scheme by Liu *et al.* adopted *one-transistor-one-memristor* (1T1M) cell structure. The use of selective transistor is highly recommended in data storage structure, to alleviate the sneak path leakage problem in crossbar [18]. However, it significantly enlarges the cell size ($\geq 6F^2$). Very importantly, more than 4% loss in computing accuracy has been induced by selective transistor after including its state resistance into consideration [16].

IV. The Proposed Circuit Design

We propose to develop memristor crossbar based computing engine by integrating a current sensing scheme. Instead of connecting each BL to an IFC, a current amplifier is used to detect BL current. Thus, the voltage of BL will be clamped to a fixed voltage level and the matrix-vector operation can closely follow the linear function of Eq. (2), without being affected much by the resistance distribution in memristor crossbar.

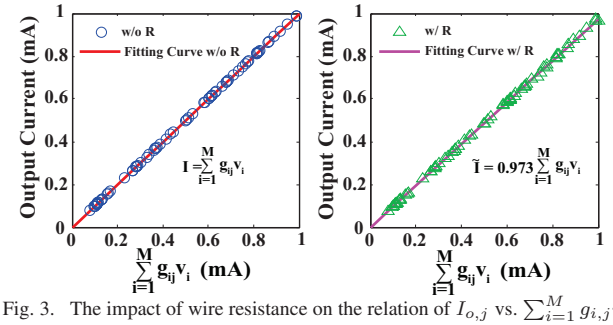

Fig. 3. The impact of wire resistance on the relation of $I_{o,j}$ vs. $\sum_{i=1}^{M} g_{i,j}v_i$.

When implementing the proposed matrix computation engine, the non-ideal effects are mainly contributed by the memristor crossbar and the current amplifier. We investigated the computation engine design made of a 144×144 memristor crossbar and the current amplifier at 130nm technology node. In this section, we will describe the detailed design considerations and analyze the computation accuracy loss.

A. Memristor Crossbar Array

It has been well known that the sneak path leakage is a major concern in memristor crossbar design [18][22]. The sneak path leakage refers to the unexpected parasitic current leakage caused by those unselected cells. To solve the issue, *one-transistor-one-memristor* (1T1M) and *one-selector-one-memristor* (1S1M) cell structures have been investigated when using crossbar arrays for data storage [23][24] and neuromorphic systems [16].

Our approach conducts the computation in a parallel mode: all inputs are represented by analog voltage signals and sent to WLs of crossbar array simultaneously. The current amplifier help keep BL at a constant voltage level. In this way, all the cells are selected and accessed at the same time. The impact of sneak path leakage is negligible in such a *multiple inputs multiple output* (MIMO) operation [25] so the BL current can follow Eq. (2). Therefore, we are able to adopt the memristor-only cell structure in this work which offers the minimal cell size of $4F^2$ while assuring computation accuracy.

The series wire resistance could also affect the BL current and therefor distort the realization of Eq. (2). To quantitatively evaluate the impact, we compared the relations of BL current ($I_{o,j}$) and $\sum_{i=1}^{M} g_{i,j}v_i$, with and without including the wire resistance into considerations. Each configuration conducted 2,000 simulations with randomly generated input data. Fig. 3 summarizes the simulation results, where only a small subset of data is included for better illustration.

In the simulations, we assumed 3-bit resistance states of memristor device, that is, eight resistance levels from $\text{R}_{\text{on}} = 50\text{K}\Omega$ to $\text{R}_{\text{off}} = 1\text{M}\Omega$. The resistance patterns of crossbar were randomly picked which cover the major portion of the range of $\sum_{i=1}^{M} g_{i,j}v_i$. For the 130nm technology adopted in the work, the wire resistance per cell is about 0.52Ω. The simulation results show that when the wire resistance is not included, all the output currents strictly follow the theoretical analysis in Eq. (2) which is not affected by data patterns. A small shift occurs after including the wire resistance into the simulation. Even though, the linearity between the output current and $\sum_{i=1}^{M} g_{i,j}v_i$ can still maintain because of the large resistance value of memristor. The fitting curve obtained in our implementation is

$$\tilde{I}_{o,j} = \gamma I_{o,j}, \tag{3}$$

where $\gamma = 0.973$. Note that our result is consistent to [26], where the largest reading error is less than 5%. It was obtained at the farthest cell in the crossbar when all the remaining cells have the lowest resistance value of R_{on}.

B. Current Amplifier

Fig. 4 depicts the schematic of our current amplifier design. It is used to detect the output current from crossbar array. Since the conductance of memristor can represent a positive value, we can subtract the results from two crossbar arrays to obtain the computation with negative matrix elements [12]. To support

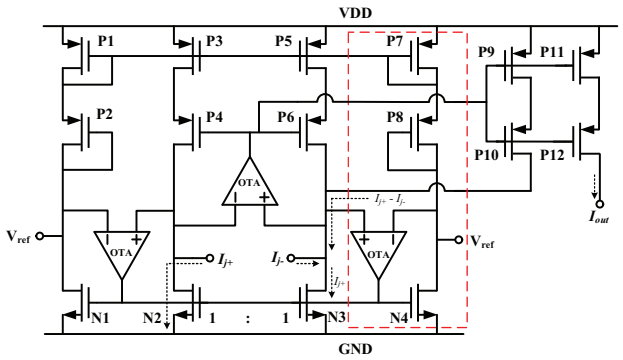

Fig. 4. The current amplifier design.

the feature, we designed a bi-lateral current buffer amplifier with two input ports I_{j+} and I_{j-}, denoting the BL currents from the crossbars for the positive and negative elements, respectively. A system-level illustration is demonstrated in Fig. 6, which shall be explained in Section V.

In this current amplifier design, V_{ref} is a reference voltage providing DC operating point. Three high-gain *operational impedance amplifiers* (OTAs) are used to clamp the voltage level of the two input ports I_{j+} and I_{j-} at V_{ref} during operation. OTAs also assist the function of the associated current mirrors, *e.g.*, keeping the same V_{ds} for $\text{N1} \sim \text{N4}$.

The input current I_{j+} injected into N2 is duplicated to N3. The subtraction of $I_{j+} - I_{j-}$ can be compensated by the branch of $\text{P9} \sim \text{P10}$. A cascode structure is adopted at the output stage to improve the accuracy of current mirror ($\text{P9} \sim \text{P12}$). For the design requiring only unilateral current input, the current amplifier corresponding to I_{j-} can be trimmed by removing the part within the red dashed box in Fig. 4.

Compared to other current amplifiers such as [27], our approach tends to minimize the output voltage variation under different amount of BL currents. Recall that in the spiking based design, a BL is directly connected to IFC, and the integration/firing operations are realized through charging/discharging a capacitor inside IFC [16]. A stable charging voltage at the capacitor is necessary in order to maintaining a fixed charging rate and therefore a constant spiking generation frequency. Our current amplifier was designed for the purpose. We evaluated the performance of I_{out} of current amplifier by fixing $I_{j-} = 0$ and sweeping I_{j+}. Here, V_{ref} is set to 200mV. The result in Fig. 5 shows that I_{out} follows well with I_{in}.

C. Overall Performance

After combining the non-ideal factors of the memristor crossbar and the current amplifier design, the output signals can be fitted by a first-order function such as

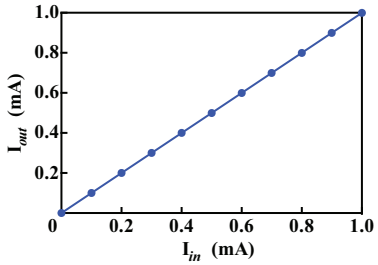

Fig. 5. The characteristic of the current amplifier: I_{in} vs. I_{out}.

$$I^*_{out} = \gamma \sum_{i=1}^{M} g_{i,j} v_i. \quad (4)$$

The root mean square errors of I^*_{out} is 2.7% .

The slight error of our design can be mitigated by many system-level designs. For example, in developing neural network models, this error can be included by substituting Eq. (4) into the training procedure so the computation accuracy at system level will not be affected much. In Section V, we will use a simple neural network to evaluate the performance of the proposed computing engine.

V. Application and Evaluation

We designed a neuromorphic system by using memristor crossbar with the proposed current sensing scheme. For each memristor crossbar, analog voltage signals are used as input data. The current generated at BL will go through a current amplifier and an IFC in sequence to produce output spikes. The performance and accuracy were evaluated through a three-layer neural network for MNIST handwritten digit recognition [19].

A. Neuromorphic System Implementation

As discussed in Section IV-B, the proposed current amplifier is used to sense out the BL current (I_{out}) with a close to ideal linearity, representing the sum of weighted multiplication. In real applications, a current output need to be transformed into a voltage signal. The spiking based architecture by Liu *et al.* [16] demonstrated a high power efficiency by eliminating the use of analog components like *analog-to-digital converters* (ADCs) and *digital-to-analog converters* (DACs). By leveraging the integrate-and-fire design concept, an approach of feeding I_{out} of the current amplifier to an IFC and therefore transferring the computation results into a digitalized format has been adopted in our design.

Fig. 6 demonstrates the system-level approach for neural network implementation. Instead of using a chain of spikes as the data input, analog voltage signals generated by DAC will be simultaneously supplied to the wordlines of crossbar to represent input vector. As discussed previously, this approach can produce more output spikes within the given computing period, offering better computation accuracy, compared to the original spike-based design. For the analog voltage inputs with different voltage amplitudes, the current amplifiers performs like buffers that isolate the crossbar array from IFCs. It helps eliminate sneak-path leakage and grantee good computing accuracy as discussed in IV-C.

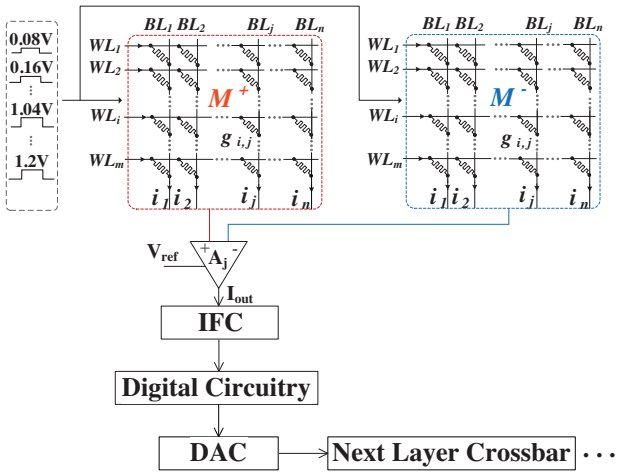

Fig. 6. The system architecture used for neural network implementation.

As shown in Fig. 6, two memristor crossbars are used in each layer, in order to realize both the positive and negative matrices terms without modifying original neural network models. The two crossbars are respectively denoted as M^+ and M^- [12]. To conduct a subtraction operation and match the mathematical algorithm in training and recall processes, the current outputs from two corresponding BLs of the two crossbars will be connected to the two input ports of an current amplifier. The computation output will be then transferred to the output of the current amplifier I_{out}, which is the input of IFC for output spike generation. The output spikes can be encoded to digital signals by digital circuitry and gives out digital voltage signal [16]. When implementing a multi-layer neural network system, the output digital signals of one layer are transferred to analog format and fed to the following layer.

In this work, we evaluated the proposed new design denoted as ***AnalogV*** in the following context and compare it with spiking based design in [16], which is denoted as ***Spiking***. Comparing the two type of system implementations, *AnalogV* offers an analog-digital flow, while *Spiking* provides a completed digitalized data transmission. The computation accuracy of *Spiking* is affected by the non-linearity effect shown in Fig. 2(a) and the selective transistor resistance in the 1T1M cell structure. The low output spiking rate in a certain computing period due to the non-linearity resulted by IFC charging and discharging described in [16] and the spiking input with 50% utilization rate are also major concerns. While, the system *AnalogV* proposed in this work can provide a higher output spiking rate because of the parallel analog voltage signal adopted in this design. Meanwhile, the crossbar of *AnalogV* is denser for it composed of only memristors. However, the design requires extra components, including current amplifier and DACs. The induced overheads have been carefully considered in the following evaluations.

B. Application of Digital Pattern Recognition

We tested and compared the two system designs on a three-layer feed-forward neural network. Here, 60,000 digital patterns from MNIST [19] were used for training, and a test set of 10,000 examples was selected randomly. During the training and testing, we resized digital patterns in 28×28 pixels from MNIST database to smaller patterns in 12×12 pixels to match the maximal allowable 144×144 memristor crossbar in this work. Each memristor represents a 3-bit data (8 resistance levels). Traditional back-propagation and delta rule were used for training and then each memristor cell in the crossbar was programmed to a target resistance [25]. Fig. 7

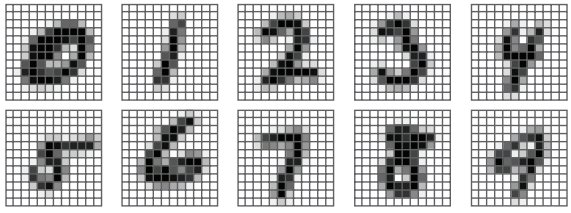

Fig. 7. An example of patterns $0 \sim 9$ with 4-bit gray scale in testing.

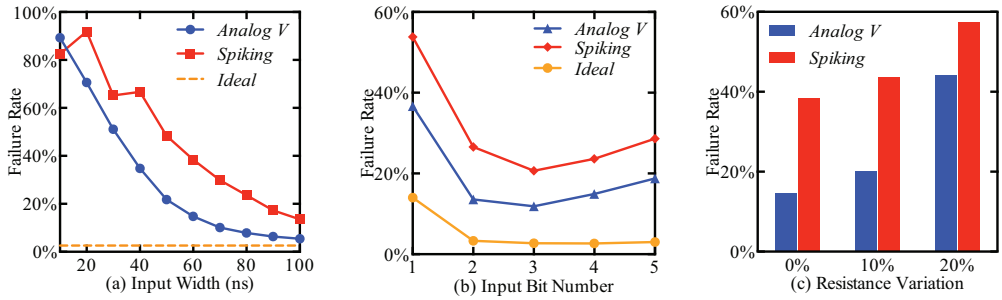

Fig. 8. The MNIST recognition failure rates of ***AnalogV*** and ***Spiking*** system designs when applying different (a) input widths, (b) input bits, and (c) resistance value variations.

shows an example of patterns $0 \sim 9$ with 4-bit gray-scale used in our training and testing.

The crossbar sizes of layer $1 \sim 3$ are 144×128, 64×128 and 64×20, respectively. The numbers were determined for the best recognition accuracy obtained at software level. We evaluated and compared the performance of *AnalogV* and *Spiking* systems under different design considerations, including the input width, input bits, and resistance value variance. The impacts of matrix-vector computation accuracy on these systems were measured by the failure rates in pattern recognition testing.

C. Design Parameter Considerations

In both system implementations, integrate-and-fire circuits are adopted to transform the current computation results to readable digital voltage signals. As such, the computing accuracy of the integrate-and-fire circuits that is mainly affected by the computation period is a major concern in the failure rate of the two systems in pattern recognition. The computation period of the IFC will be determined by both input signal width and input bits. Therefore, the input signal width and input bits will be two important design parameters needed to be consideration. As discussed above, DACs are used in our proposed system *AnalogV* and its resolution will affect the computation accuracy of the system. In the hardware design, DAC with half-bit resolution error was implemented and the induced computation accuracy loss was considered in the following evaluation. Moreover, during programming memristor with multiple value, variations in resistance value cannot be ignored and will be considered.

1) Input Width Dependence: We first tested the pattern recognizing failure rated under different input signal widths. Inputs were set to 4-bit value in this evaluation. Fig. 8(a) demonstrates the result when varying the input width from 10ns to 100ns. The failure rate under the ideal condition is 2.66%, which was obtained from software simulation without including any real implementation consideration. The simulation results showed that both *AnalogV* and *Spiking* systems are sensitive to the input width, and the new scheme *AnalogV* has the higher computation accuracy. More specific, the failure rate of *AnalogV* is 5.43% at the input width of 100ns, which is 8.11% less than that of *Spiking* design (13.54%).

The large input width dependency is mainly caused by IFC design so computation accuracy improves as the input width increases. Accordingly, the failure rate decrease as the input width grows up, especially when it is smaller than 60ns. *Spiking* demonstrates a much higher failure rate (38.43%) than *AnalogV* (14.75%) under the input width of 60ns. This is because sparse input pulses are given as input for *Spiking* and the real computation time is decrease by half. More output spikes will be generated in system *AnalogV* in a certain input width as analog signals are applied to crossbar with 100% of utilization. The failure rate fluctuation of *Spiking* at small input width indicate the high instability of the system and high randomness of the testing results. The accuracy loss caused by the non-linearity of IFC in *AnalogV* is much smaller than the accuracy decrease in *Spiking* in the same computation speed. In the other words, our proposed system *AnalogV* could obtain much higher computing speed (about twice) than *Spiking* when they target at the same computation accuracy.

Another concern on *AnalogV* design is the frequency of DACs that generate discrete analog signals as crossbar inputs. In our implementation, DAC was designed with a maximum frequency of 100MHz, corresponding to 10ns analog input width. The maximum frequency can be increased in real design but power and area will be scarified. The result in Fig. 8(a) indicates that the input width must be longer than 60ns in order to achieve a reasonable failure rate. Therefore, the DAC with 100MHz frequency is large enough and will not be a constraint in the system design.

2) Input Bits Dependence: Fig. 8(b) shows the impact of pattern input bits, reflecting the pattern color depth on the system failure rate. Based on the above analysis, we evaluated the input bits dependence by using 60ns input width. The two systems were first tuned till they can obtain similar accuracy. Comparing with prior design *Spiking*, *AnalogV* obtained lower failure rate when the input pixel has more than 2 bits It matches well with the simulation results discussed above. As can be seen from the ideal curve, there is a valley value. This is because more input bits per pixel of input images indicates higher complexity of neuromorphic system should afford, and the computation accuracy is limited by the available resistance states of memristors. Practically, we observed that the values of $\sum_{i=1}^{M} g_{i,j} \cdot v_i$ drops statistically, leading to higher quantification error in IFC. These two opposite factors together influence the trend shown in Fig. 8(b). We chose 4-bit color-depth to obtain a relatively high accuracy in difference systems.

3) Impact of Resistance Value Variation: The impact of the resistance value variation in computation failure rate was evaluated too. The failure rates of both systems with 0%, 10%, and 20% resistance variation are summarized in Fig. 8(c). The systems failure rate increases with the increasing of resistance value variation. System *Spiking* is more vulnerable to the resistance value variation because of the lower output spike number in it. We observe a large failure rate increase when increasing the resistance value variation from 10% to 20%.

TABLE I. SYSTEM PERFORMANCE COMPARISON

	Area	Power	Speed	Energy
AnalogV (This work)	0.426mm^2	100.6mW	16.7MHz	6.04nJ
Spiking [16]	0.476mm^2	82.93mW	10MHz	8.29nJ
Difference	-10.5%	+17.6%	+40.1%	-27.1%

The results show that the failure rate of our proposed design *AnalogV* is under 20% when the variation is 10%, while the failure rate of *Spiking* increases to more than 45%.

D. Design Comparison

The area, power, speed and energy consumption of the two systems were compared and summarized in Table I. Area and power data are based on the systems implementation at 130nm technology node. The DACs and the current amplifiers used in *AnalogV* induce more area and power consumption. However, less IFCs and digital circuits are needed in *AnalogV* as the subtraction of computing result from the positive and negative crossbars can be executed by current amplifier. The 1T1M crossbar structure of *Spiking* induces more area cost.

Comparing to *Spiking*, *AnalogV* obtains 10.5% decrease in area while increases power consumption 17.6%. This is caused by the extra power consumed by the current amplifier and DACs. When testing the speed of *AnalogV* and *Spiking*, we selected 60ns and 100ns as input pulse widths respectively to maintain an approximate similar failure rate 15%. *AnalogV* executes much faster than *Spiking*, obtaining 40.1% speed improvement. Overall, *AnalogV* lowers the energy consumption 27.1% comparing with *Spiking*.

VI. CONCLUSIONS

In this work, a memristor crossbar based computing engine using current sensing was proposed for matrix-vector computation. The parallel exaction was realized by applying analog voltages to every wordline of memristor crossbar. A current amplifier circuit was designed which mirrors the current from the crossbar with a slight accuracy degradation. To evaluate the computation accuracy of the scheme, we implemented a three-layer feed-forward neural network and plugged different memristor crossbar based computing engines for comparison. We thoroughly analyzed the newly proposed system and a prior reported design from perspectives of accuracy, speed, area, and energy. The results show that our system has lower area and energy consumption with a higher speed than prior spiking based design. The computation accuracy of the new system based on our computing engine can reach 94.6%. Overall, it demonstrates a good computation accuracy in matrix computation with a lower energy consumption.

ACKNOWLEDGMENT

This work was supported in part by AFRL FA8750-15-2-0048, 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 grant agencies or their contractors.

REFERENCES

[1] D. Evans and Hatzopoulos, "A parallel linear system solver," *International Journal of Computer Mathematics*, vol. 7, no. 1, pp. 227–238, 1979.

[2] P. Simon, *Too Big to Ignore: The Business Case for Big Data*. Wiley, 2013.

[3] V. Kelefouras *et al.*, "A methodology for speeding up matrix vector multiplication for single/multi-core architectures," *The Journal of Supercomputing*, vol. 71, pp. 2644–2667, July 2015.

[4] S. M. Qasim *et al.*, "FPGA design and implementation of dense matrix-vector multiplication for image processing application," in *IJCSNS*, October 2010.

[5] P. Saha *et al.*, "Improved matrix multiplier design for high-speed digital signal processing applications," *IET Circuits, Devices and Systems*, vol. 8, no. 1, pp. 27–37, Jan 2014.

[6] X. Zhang *et al.*, "Exploring potentials of perpendicular magnetic anisotropy stt-mram for cache design," in *Solid-State and Integrated Circuit Technology (ICSICT), 2014 12th IEEE International Conference on*, Oct 2014, pp. 1–3.

[7] L. O. Chua, "Memristor-the missing circuit element," in *IEEE Transactions on Circuit Theory*, vol. 18, September 1971, pp. 507–519.

[8] D. B. Strukov *et al.*, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[9] K.-H. Kim *et al.*, "A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications," *Nano letters*, vol. 12, no. 1, pp. 389–395, 2011.

[10] F. Clermidy *et al.*, "Advanced technologies for brain-inspired computing," in *ASP-DAC*, Jan 2014, pp. 563–569.

[11] G. S. Rose *et al.*, "Leveraging memristive systems in the construction of digital logic circuits," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 2033–2049, June 2012.

[12] M. Hu *et al.*, "Hardware realization of BSB recall function using memristor crossbar arrays," in *DAC*. ACM, 2012, pp. 498–503.

[13] S. H. Jo *et al.*, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[14] M. Sharad *et al.*, "Ultra low power associative computing with spin neurons and resistive crossbar memory," in *DAC*. ACM, 2013, pp. 107:1–107:6.

[15] M. Prezioso *et al.*, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, pp. 61–64, 2015.

[16] C. Liu *et al.*, "A spiking neuromorphic design with resistive crossbar," in *DAC*. ACM, July 2015, pp. 1–6.

[17] J. Yang *et al.*, "Memristive devices for computing," *Nature Nanotechnology*, vol. 8, no. 1, pp. 13–24, Jan 2013.

[18] C. Liu and H. Li, "A weighted sensing scheme for ReRAM-based cross-point memory array," in *ISVLSI*, 2014, pp. 65–70.

[19] Y. LeCun *et al.* The MNIST DATABASE of handwritten digits.

[20] S. Yu *et al.*, "Investigating the switching dynamics and multilevel capability of bipolar metal oxide resistive switching memory," *Applied Physics Letters*, vol. 98, no. 10, p. 103514, 2011.

[21] B. Li *et al.*, "Merging the interface: Power, area and accuracy co-optimization for rram crossbar-based mixed-signal computing system," in *DAC*. ACM, June 2015, pp. 1–6.

[22] A. Flocke and T. Noll, "Fundamental analysis of resistive nano-crossbars for the use in hybrid Nano/CMOS-memory," in *Solid State Circuits Conference*, Sept 2007, pp. 328–331.

[23] J.-J. Huang *et al.*, "One selector-one resistor (1S1R) crossbar array for high-density flexible memory applications," in *Electron Devices Meeting*, Dec 2011, pp. 31.7.1–31.7.4.

[24] S.-S. Sheu *et al.*, "A 5ns fast write multi-level non-volatile 1 K bits RRAM memory with advance write scheme," in *VLSI Circuits*, June 2009, pp. 82–83.

[25] M. Hu *et al.*, "Memristor crossbar-based neuromorphic computing system: A case study," *TNNLS*, vol. 25, no. 10, pp. 1864–1878, Oct 2014.

[26] D. Roclin *et al.*, "Sneak paths effects in cbram memristive devices arrays for spiking neural networks," in *Nanoscale Architectures*. ACM, July 2014, pp. 13–18.

[27] C. Y. Leung *et al.*, "An integrated cmos current-sensing circuit for low-voltage current-mode buck regulator," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52:7, July 2005.