MNSIM: Simulation Platform for Memristor-based Neuromorphic Computing System

Lixue Xia[1], Boxun Li[1], Tianqi Tang[1], Peng Gu[12], Xiling Yin[1], Wenqin Huangfu[1], Pai-Yu Chen[3], Shimeng Yu[3], Yu Cao[3], Yu Wang[1], Yuan Xie[2] and Huazhong Yang[1]
[1]Dept. of E.E., Tsinghua National Laboratory for Information Science and Technology (TNList), Tsinghua University, Beijing, China
[2] Department of Electrical and Computer Engineering, University of California at Santa Barbara, California, USA
[3]School of Electrical, Computer and Energy Engineering, Arizona State University, Arizona, USA

***Abstract*—Memristor-based neuromorphic computing system provides a promising solution to significantly boost the power efficiency of computing system. Memristor-based neuromorphic computing system has a wide range of design choices, such as the various memristor crossbar cell designs and different parallelism degrees of peripheral circuits. However, a memristor-based neuromorphic computing system simulator, which is able to model the system and realize an early-stage design space exploration, is still missing. In this paper, we develop a memristor-based neuromorphic system simulation platform (MNSIM). MNSIM proposes a general hierarchical structure for memristor-based neuromophic computing system, and provides flexible interface for users to customize the design. MNSIM also provides a detailed reference design for large-scale applications. MNSIM embeds estimation models of area, power, and latency to simulate the performance of system. To estimate the computing accuracy, MNSIM proposes a behavior-level model between computing error rate and crossbar design parameters considering the influence of interconnect lines and non-ideal device factors. The error rate between our accuracy model and SPICE simulation result is less than 1%. Experimental results show that MNSIM achieves more than 7000 times speed-up compared with SPICE and obtains reasonable accuracy. MNSIM can further estimate the trade-off between computing accuracy, energy, latency, and area among different designs for optimization.**

I. Introduction

The explosion of data amount brings higher demand for power efficiency in modern computing system design [1]. However, it becomes more and more difficult to achieve substantial power efficiency gains directly through the scaling down of traditional CMOS technique. Meanwhile, the memory bandwidth required by high-performance CPUs also meets an ever-increasing "memory wall" challenge to the efficiency of von Neumann architecture [2]. Consequently, there is a growing research interest of exploring emerging nano-devices and new computing architectures to further improve power efficiency [3], [4].

In recent years, the innovation of memristor and memristor-based computing system provides a promising solution to significantly boost the power efficiency of computing systems. The nonvolatile property and the crossbar structure provide a promising alternative to the non-von Neumann architecture by changing the architecture to merge computation and memory [5]. By taking advantage of these characteristics, researchers have designed many neuromorphic systems using memristor crossbar and get great efficiency gains [6].

As the ultimate goal of memristor-based neuromorphic system research is to implement large-scale neuromorphic applications on memristor, researchers need to estimate and optimize the performance of their designs before fabrication. However, none of the existing system simulation platforms can completely support the simulation of memristor-based computing system. Traditional architecture simulator like GEM5 [7] can not simulate memristor device and memritor-based computing structure. NVSim [8] and NVMain [9] are memory-oriented simulators which can simulate and optimize the designs of memristor-based memory. But the peripheral circuit structure of NVSim/NVMain is fixed for memory simulation, so they can not support the special operations of computing system. Researchers now can only simulate the system with circuit-level simulators such as SPICE by using memristor models, or use other circuit-level simulator that has embedded memristor models like NVMspice [10]. However, the simulation time obviously increases when the scale of network gets larger. As a result, a fast simulation platform oriented to memristor-based computing system with an accurate behavior-level model is highly demanded.

However, there are some challenges to develop a behavior-level simulation platform for memristor-based neuromorphic system. First, since the optimal circuit of memristor-based neuromorphic system is far from conclusive, a simulation platform needs to be general and scalable enough to support the various designs. Therefore, how to propose a flexible architecture of memristor-based neuromorphic system is the major challenge for simulator design. Second, computing accuracy is the main metric that needs to be estimated in a memristor-based computing system, but a behavior-level estimation model is still missing. Third, there are lots of design parameters can be adjusted in the circuit design, so the behavior-level simulation platform must support the design space exploration with a fast simulation speed.

This work proposes MNSIM, the first behavior-level simulation platform for the memristor-based neuromorphic computing system. The main contributions of MNSIM are as follows:

1) We propose a general hierarchical structure of memristor-based neuromophic computing system. MNSIM provides flexible interface for users to customize the design through this scalable architecture. MNSIM also provides a reference design for large-scale neuromorphic system simulation.
2) MNSIM proposes a behavior-level model of memristor-based computing accuracy, and embeds the estimation model of area, power, and latency. The behavior-level model can accelerate the estimation more than 7000× compared with SPICE.
3) MNSIM can explore the huge design space of memristor-based neuromorphic computing systems and give the optimal design with details, which can guide the design of memristor-based neuromorphic computing systems.

II. Preliminaries

A. Memristor Device

A memristor device is a passive two-port element with variable resistance states. There are multiple kinds of devices which can be used as memristor, such as Resistive Random Access Memory (RRAM), Phase Change Memory (PCM), and etc. The memristor devices can be used to build the crossbar structure as shown in Fig. 1(f). If we store the "matrix" by the conductivity of the memristor device in crossbar ($g_{K,J}$) and input the voltage "vector" signals, the memristor crossbar is able to perform analog matrix-vector multiplication. The relationship between the input voltage vector and output voltage vector can be expressed as follows [11]:

$$v_{out,k} = \sum_{j=1}^{N} c_{k,j} \cdot v_{in,j} \quad (1)$$

This work was supported by 973 Project 2013CB329000, National Natural Science Foundation of China (No. 61373026, 61261160501), Brain Inspired Computing Research, Tsinghua University (20141080934), Tsinghua University Initiative Scientific Research Program, the Importation and Development of HighCaliber Talents Project of Beijing Municipal Institutions.

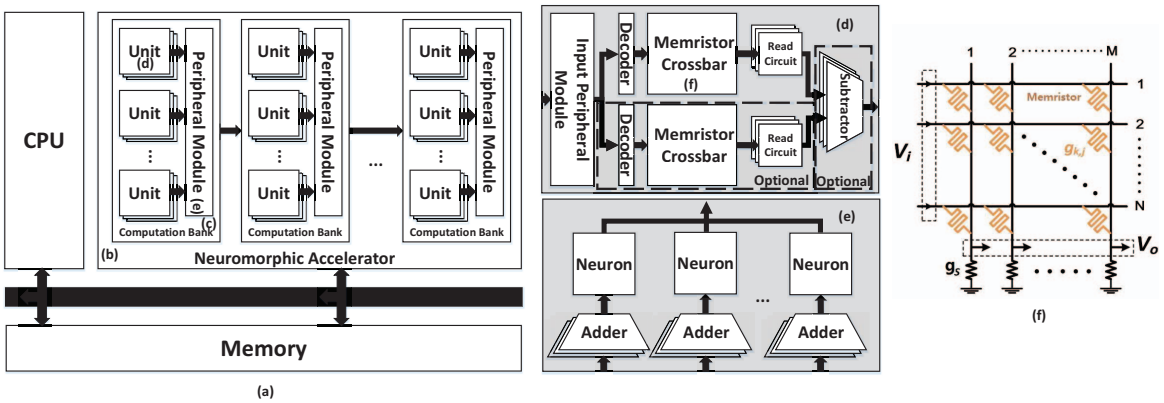


Fig. 1. (a) The overview architecture. (b) Hierarchical Structure of Memritor-based Neuromorphic Computing System. (c) Computation Bank. (d) Computation Unit, where the circuits in dotted blocks are optional to implement bipolar weight. (e) Peripheral Module. (f) Memristor Crossbar.

where $v_{in,j}$ is the jth element of input voltage vector denoted by j ($j = 1, 2, ..., N$), $v_{out,k}$ is the kth element of output voltage vector denoted by k ($k = 1, 2, ..., N$), and $c_{k,j}$ is the matrix data. The matrix data can be represented by the conductivity of the memristor device and the load resistor (g_s) as:

$$c_{k,j} = \frac{g_{k,j}}{g_s + \sum_{l=1}^{N} g_{k,l}} \tag{2}$$

As a result, analog ANNs and other neuromophic systems can be implemented by memristor-based structure [12].

B. Comparison of Memristor-based Memory and Computing System

Researchers have implemented kinds of non-volatile memory using memristor devices, but the memory-oriented design can not be directly used to process computation for two main reasons. First, to utilize the advantage of crossbar structure as shown in Fig. 1(f), all the memristor cells in a crossbar need to be selected when processing matrix-vector multiplication, but in memory we only need to select a single cell in a crossbar. The cell selection scheme needs to be refined, and the corresponding circuits need to be adjusted. Second, the neuron function circuits (i.e. sigmoid function in ANN) need to be embedded into the structure when implementing a complete neuromorphic computing system.

III. Memristor-based Neuromorphic Computing Structure

A. The Hierarchical Structure Analysis of Memeristor-based Neuromorphic Computing System

To simulate various system designs while remaining the similar simulation architecture at the same time, it is necessary to propose a general and flexible structure for memristor-based computing system. Memristor-based computing system serves as an acceleration platform which obtains data from CPU or memory, as shown in Fig. 1(a). Generally, the neuromorphic algorithm consists of multiple cascaded network layers, and the functions of layers are similar. Therefore, the whole neuromorphic system can be divided into multiple cascaded modules with similar functions, where each module represents the operation of a network layer. This module is named **Computation Bank** in MNSIM.

When the scale of a network layer is large, like a 2500×2000 network layer in ANN [13], the amount of cells can be more than 5×10^6. However, the existing technology can not fabricate a single memristor crossbar containing over 10^6 cells. Moreover, as the size of crossbar grows, the non-ideal factors of memristor crossbar become serious [14], which obviously reduces the computing accuracy. Therefore, to process large-scale neuromorphic application, the computation bank must further contains multiple individual units with a peripheral module to combine the results of units. MNSIM calls these individual units **Computation Unit**s.

As a result, most of the circuit designs of memristor-based neuromorphic system can be represented by the hierarchical structure containing 3 levels: system level, bank level, and unit level. The hierarchical structure is shown in Fig. 1(b). Based on this abstraction, MNSIM provides flexible customization interface while sustain the system architecture unchanged. As shown in Table I, users can configure the design from different level to process the simulation of various neuromorphic systems. In addition, MNSIM supposes that all the weight matrix can be completely stored in the multiple memristor crossbars. This is because that high integration is the main advantage of memristor-based structure, and storing all weights without repeatedly writing can better utilize the energy efficiency of memristor. But the structure can also be adjusted for reconfigurable designs as discussed in Section III-E.

The detailed reference design of each level is discussed below. Moreover, if users need to use some special designs that are not listed in Table I, they can adjust the detailed design of corresponding layer through a friendly C++ interface.

B. Computation Bank

A computation bank processes the computation of a single network layer, and multiple computation banks cascaded into a whole neuromorphic system. Each computation bank consists of multiple computation units and a **Peripheral Module**. When using multiple crossbars to implement the matrix-vector multiplication, we actually divide the matrix into many blocks, and then merge the results of multiple small matrix-vector multiplications of a row by an addition operation as:

$$\vec{V_{out,j}} = \sum_{k=1}^{N} W_{k,j} \vec{V_{in,k}} \tag{3}$$

where $\vec{V_{in,k}}$ is the input data vector in kth row (i.e. $\vec{V_{in,1}} = (v_{in,1}, v_{in,2}, ..., v_{in,s})$ if the crossbar size is s), $\vec{V_{out,j}}$ is the output data vector in jth row, and $W_{k,j}$ is the unit's weight matrix in kth row and jth column. Each unit processes a block of the matrix-vector multiplication, and the peripheral module merges the computation results of multiple units. The reference structure of computation bank is shown in Fig. 1(c).

In addition, considering that the digital signal has a better potential to achieve noiseless communication of multiple units for large network, MNSIM embeds digital input/output design of computation unit and peripheral module as a reference design. If users want to use the design with analog communication, they can move the read circuits from units to peripheral modules to process the simulation.

C. Peripheral Module

The peripheral module contains the adders and neurons, as shown in Fig. 1(e). The adders are used to merge the multiple individual matrix-vector multiplication results, while neuron is the module to provide non-linear characteristic in neuromorphic algorithms, like the

sigmoid function in ANN. Since the neuron function is non-linear, so it cannot be separately operated before the adding operation in Eq. (3). So, the neuron function can only be implemented outside the units, and operates after the results of different units have been added together.

D. Computation Unit

A computation unit consists of five main modules: memristor crossbar, address decoder, read circuit, input peripheral circuit, and control module. The reference structure of computation unit provided in MNSIM is shown in Fig. 1(d).

a) Memristor Crossbar: is the main part of the computation unit which accomplishes the memory and computation functions of neuromorphic system. Specifically, since the resistance of memristor devices can only be positive, a unit needs to contain two memristor crossbars and uses the difference value of the corresponding cells to reflect the bipolar weight for some applications, as shown in Fig. 1(d). The polarity of network weight is configurable, and the $crossbar\ size$ can be adjusted for design space exploration.

In default, MNSIM uses the RRAM model from Stanford University as the device model [15], and also maintain the scalability for other memristor devices. During the read and compute operations, the difference between RRAM and other memristor devices is just the resistance range, so the operation mechanism and circuit structure of other devices are the same as RRAM. In addition, both the MOS-accessed cell (1T1R) and the Cross-point cell (0T1R) are modeled in MNSIM.

b) Address Decoder: is the module to select specific column/row through a transfer gate. Only one cell needs to be operated in write operation, so two decoders are needed for each crossbar to select the specific row and column. Specifically, the decoder design is different from traditional decoder used in memory, because we need to select all the input ports of crossbar. A computation-oriented crossbar decoder is designed to solve this problem, which is introduced in Section IV-B2. The detailed decoder circuit is determined by $crossbar\ size$.

c) Input Peripheral Circuit: is the module generating the signal (pulse or voltage) to be transmitted into crossbar, which contains DACs and transfer gates. Memristor devices usually use pulse signals for writing, reading, and computing. To implement voltage pulses with various amplitudes, MNSIM inputs the analog voltage data (converted by DACs) into a transfer gate which is controlled by digital pulses.

d) Read Circuit: is the module processing the signal coming from the memristor crossbar. The read circuit can be ADCs or multilevel Sensing Amplifiers (SAs). If the weights of network are bipolar, extra subtractors are needed to merge the signals comes from two crossbars. The amount of read circuits can be adjusted if the users want to only compute p columns' results and sequentially process t times to obtain complete results. MNSIM can use the parallelism degree p as a variable to estimate the trade-off between latency and area.

e) Control Module: is the module generating the control signals and leading the operation of reminder modules. An important function of control module is to implement the switch policy between write and verify operation when using pulse signal to write a weight into the memristor cell [16].

E. Customized Design

Users can customize the design in multiple levels. From bank level, users can customize the connection between units. For example, if users wants to generate reconfigurable network instead of a fixed cascaded structure, they can distribute the peripheral module into each unit to obtain a system only consists of reconfigurable units. This design will introduce lots of redundant circuits, so it is not chosen as the reference structure in MNSIM. From unit level, the detailed structure of computation units and peripheral modules can be adjusted. For example, the structure proposed in [17] can eliminate the DACs and ADCs (or multilevel SAs) around crossbar, so the users can remove the corresponding modules of unit to simulate this structure. In addition, the detailed estimation model of each circuit module is also flexible. For example, if users want to use a special kind of DAC, they only need to change the performance estimation model of corresponding circuits.

TABLE I
THE CONFIGURATION LIST OF MNSIM

Inputs	Level	Default Value	Comment
Application	System	ANN	Algorithm Type
Application_Scale	System	-	Layers of Application
Weight_Bit	System	8	Weight Precison
Input_Bit	System	8	Signal Precison
Network_Scale	Bank	-	Scale of Each Layer
Weight_Polarity	Unit	2	
CMOS_Tech	Unit	90nm	
Cell_Type	Unit	1T1R	1T1R/0T1R
Memristor_Model	Unit	RRAM	
Interconnect_Tech	Unit	28nm	
Crossbar_Size	Unit	128	
Parallelism_Degree	Unit	0	0 means All Parallel
Resistance_Range	Unit	[500 500k]	Min/Max Memristor Resistance

In addition, NVSim [8] is a powerful simulator for memristor-based non-volatile memory. Therefore, to fully explore the compatibility between memritor-based memory and memristor-based computing system, we provide an interface for each computation-oriented module (i.e. sigmoid circuit) to be compatible with NVSim. Users can easily introduce some NVSim results into MNSIM; or use MNSIM results in NVSim by adding some circuity models.

IV. SIMULATOR FRAMEWORK

A. MNSIM Overview

MNSIM supports the function of signal simulation, performance estimation, and design optimization. As a basic function, MNSIM can simulate the voltage, current, and practical mapped memristor resistance of neuromorphic computing system. Based on the CMOS technology data and some estimation models, MNSIM can further estimate the area, power, latency, and computing accuracy. Users need to provide the configuration file of MNSIM to determine the simulation target and other optional input parameters. The details about configuration file are shown in Table I. If the users do not determine all configurations, MNSIM can explore the design space and give the optimal design with details.

B. Area, Power, and Latency Models

MNSIM provides the following models to estimate the area, power, and latency of each module in Fig. 1(a), and also supports users to embed original devices by providing the performance parameters. Some modules have been adequately analyzed by researchers, such as ADC, DAC, and etc. Therefore, we mainly introduce the models of memristor crossbar and decoder because MNSIM refined these circuits to support memristor-based computation.

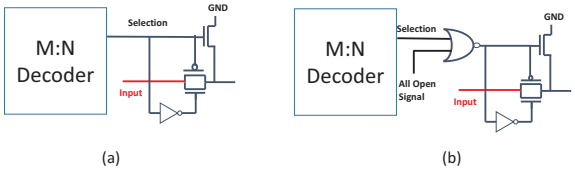
1) Memristor Crossbar: Since the crossbar structures do not need to be changed for computation, MNSIM uses the existing area model of memristor-based memory to estimate the crossbar area. The area estimation models of MOS-accessed cell and cross-point cell are [18]:

$$AREA_{cell,MOS-accessed} = 3(W/L+1)F^2 \quad (4)$$

$$AREA_{cross-point} = 4F^2 \quad (5)$$

where W/L is the technology parameter of transistor in each cell, and F is the size of memristor technology node.

However, we select all cells simultaneously in computing phase, the power consumption is larger than that of memory-oriented circuit. MNSIM uses the harmonic mean of minimum and maximum resistance of memristor to replace all cells' resistance values as the average case estimation, and uses minimum resistance to replace all cells' resistance values as a worst case estimation. Harmonic mean is chosen because both power and parallel resistance are computed using cell's resistance as a denominator.



(a) (b)

Fig. 2. (a). Memory-oriented Decoder. (b). Computation-oriented Decoder to select all cells. A NOR gate is added.

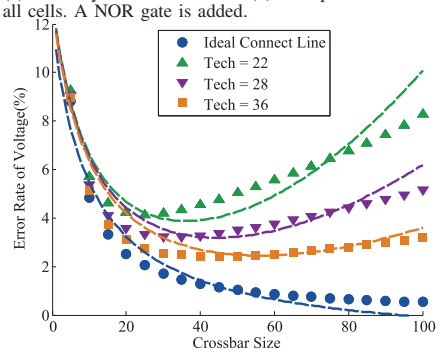


Fig. 3. The error rate fit curves of output voltages with different crossbar sizes and interconnect technology nodes. The scattered points are simulated by SPICE, and the lines reflect the proposed estimation model.

2) Decoder: The decoder of memristor crossbar needs to be modified to support the function of selecting all cells in the computing phase. Specifically, the traditional decoder used in memory is an address selector controlling the transfer gate between input signal and crossbar, as shown in Fig. 2(a). MNSIM proposes a simple but efficient method to modify this decoder into a computation-oriented decoder shown in Fig. 2(b). A NOR gate is placed between the address decoder and the transfer gate, and a control signal is connected into NOR gate. When processing computation, the control signal is at high voltage and turns on all transfer gates through NOR gate. This modification only adds a NOR gate into the structure, so MNSIM directly adds the area, power and latency cost of a NOR gate into the estimation model.

3) Other Modules: To support the estimation of various technology nodes, MNSIM provides a reference transistor-level design for each module and uses the technology parameters from CACTI, NVSim, PTM (Predictive Technology Model), and other technology documents to estimate the parameters of transistor-based modules. Especially, MNSIM uses a multilevel SA [19] as the reference read circuit design to support simulation of different data precisions. The circuit from [20] is chosen as the reference sigmoid module. MNSIM can also use the parameters of existing libraries and designs to replace the analytical models. In addition, it is difficult to directly calculate the power and latency of analog circuits only by the connection of transistors, because the DC operating points varies a lot between different transistors. When we change the technology node of transistors, we still remain the function of each transistor in these analog circuits, so the different performance between multiple technology nodes is mainly caused by the different circuit parameters of transistors (i.e. transistor size, capacitor of side wall, and On-current density). MNSIM uses the simulation results from SPICE to revise the operating points of analog transistors in order to guarantee the simulation accuracy.

C. Computing Accuracy Model

Computing accuracy is an important characteristic for a computing system. However, the relationship between the numerical output number of computing system and the classification result of application is too complex, which varies in different algorithms and different applications. Therefore in MNSIM, the computing accuracy of memristor crossbar is defined as the relative accuracy of numerical computing results between memristor-based structure and ideal CPU computation, as discussed in Section IV-C. This accuracy definition represents the computing precision of memristor circuit.

The unexpected voltage reduction caused by wire consumption and other cells is a main non-ideal factor that influences the computing results of memristor-based crossbar, and the impact increases obviously when the size of crossbar gets larger. Moreover, the V-I characteristic of memristor device is not linear in the same resistance state [14], [15], [21], which also makes computation inaccurate. MNSIM analyzes the coarse relationship between computing error rate and the crossbar circuits based on the existing research about these non-ideal factors. For a crossbar with M rows and N columns, when the input voltages equal, the output voltage of a column is:

$$V_o = V_i \times \frac{R_s}{R_{parallel} + R_s} \tag{6}$$

where $R_{parallel}$ is the parallel resistance of the whole column, and R_s is the equivalent sensing resistance. If we take the resistance of interconnect line between neighboring cells r into account, the $R_{parallel}$ is larger than the parallel resistance of memristor cells, as shown in Eq. 7. Considering that r is much less than the resistance of memristor, the difference between denominators can be ignored. In the worst case, all memristors are at the minimum resistance, and the worst column is the farthest column from input signals. So the $R_{parallel}$ can be further approximatively estimated by:

$$\frac{1}{R_{parallel}} \approx \sum_{m=1}^{M} \frac{1}{R_{m,n} + mr + nr} \approx \frac{M}{R_{min} + (M+N)r} \tag{7}$$

where $R_{m,n}$ is the resistance of memristor cell in the mth row and the nth column, and R_{min} is the minimum resistance of memristor device. Furthermore, taking the non-linear V-I characteristics into account, the practical resistance of memristor device R_{act} differs from the ideal value R_{idl}. By substituting Eq. (7) into Eq. (6), we can estimate the actual output voltage. After simplification, the fractional error of output voltage is:

$$\frac{V_{o,idl} - V_{o,act}}{V_{o,idl}} = \frac{[(M+N)r + R_{act} - R_{idl}]R_s}{[R_{act} + (M+N)r + R_s M](R_{idl} + R_s M)} \tag{8}$$

We use M, N, and r as variables to simulate the error rate of output voltages on SPICE, and fit the relationship according to Eq. (8) to fill the accuracy module, as shown in Fig. 3. The root mean squared error of this fitting curve is less than 0.01.

To further evaluate the accuracy from higher level, MNSIM transforms the above voltage error rate into the digital data deviation. Generally, the results of matrix-vector multiplication are linear quantized into k levels by the read circuits. The minimum value of the voltage signal equals 0 because the minimum value is obtained when all the input voltage signals of crossbar are 0. Therefore, given the quantization interval $V_{interval}$, the $k-1$ quantization boundaries are $\{0.5V_{interval}, ..., (k-1-0.5)V_{interval}\}$.

According to the non-ideal factor analysis, the input data of these convert circuits contains an maximum deviation rate of ϵ, which causes a read deviation when converting the analog voltage into a digital data. MNSIM uses both the worst case deviation and the average deviation to evaluate the accuracy. In the worst case, the ideal computing result signal is just around the largest quantization boundary $(k-1-0.5)V_{interval}$ and needs to be recognized as the maximum value $k-1$. Influenced by the non-ideal crossbar, the practical computing result in the worst case is $(k-1-0.5)V_{interval} \times (1-\epsilon)$, so the maximum deviation between the actual analog signal and the ideal quantization level is $[(k-1-0.5)\epsilon + 0.5]V_{interval}$. The maximum digital deviation can be calculated by:

$$MaxDigitalDeviation = \lfloor (k-1.5)\epsilon + 0.5 \rfloor \tag{9}$$

For example, when k equals 64 and ϵ equals 10%, the $MaxDigitalDeviation$ equals 6, which means that the maximum value 63 can be wrongly read as 57. Therefore, the maximum error rate is:

$$MaxErrorRate = \frac{\lfloor (k-1.5)\epsilon + 0.5 \rfloor}{k-1} \tag{10}$$

TABLE II
VALIDATION RESULTS WITH RESPECT TO A RRAM-CROSSBAR-BASED TWO LAYER ANN. THE TECHNOLOGY NODE OF CMOS IS 90NM.

Metric	MNSIM	SPICE Result	ERROR
Computation Power(mW) (Decoder+Crossbar)	17.20	16.34	+5.26%
Read Power(mW) (Decoder+Crossbar)	2.39	2.44	-2.05%
Computation Energy(uJ) (3-layer ANN)	0.525	0.487	+7.73%
Latency(ns)	381.49	405.50	-5.92%
Average Relative Accuracy(%)	95.41	94.57	**-0.89%**

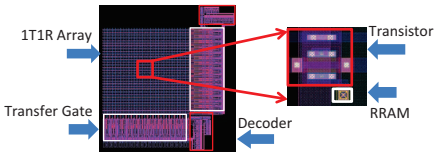


Fig. 4. The layout of 32 × 32 1T1R RRAM crossbar with the decoder in 130nm technology.

However, this maximum error rate is obtained only when the computing result is around the maximum value, so MNSIM also uses an average error rate to evaluate the computing accuracy. Similar to the above analysis, the digital deviation of a specific quantization level i can be represented by $\lfloor i\epsilon + 0.5 \rfloor$, where we use i instead of $i - 0.5$ or $i + 0.5$ to reflect the average situation. Therefore, the average deviation is:

$$AvgDigitalDeviation = \frac{\sum_{i=0}^{k-1} \lfloor i\epsilon + 0.5 \rfloor}{k} \quad (11)$$

Moreover, when simulating the application with multiple network layers, the error rate needs to be transported through crossbars, which means the input signal of next layer has a fluctuation. Suppose that the digital error rate of the previous layer is δ_{d1} while the computing error rate of the next layer's crossbar is ϵ_{c2}, the practical analog voltage result of the next layer is limited by:

$$(1 - \delta_{d1})(1 - \epsilon_{c2})V_{idl} \leq V_{act} \leq (1 + \delta_{d1})(1 + \epsilon_{c2})V_{idl} \quad (12)$$

which can be substituted into Eq. (9)˜(11) to further analyze the read error rate of next layer. MNSIM uses this propagation model to evaluate the final accuracy of the whole system layer by layer.

V. EXPERIMENTAL RESULTS

A. Validation

We choose a 3-layer ANN with two 128×128 network layers as the application to validate the power and latency module. The SPICE results are the average value of 20 random samples of weight matrixes and 100 random samples of input vectors. The technology node of CMOS is 90nm. The results are shown in Table II, where all the error rates are smaller than 10% compared with the SPICE results. As for memristor-based computing accuracy part, we use an approximate computing application, the JPEG encoding processed in a 3-layer 64×16×64 ANN, for validation. The result shows the error rate of accuracy model is less than 1%, which validates the precision of proposed behavior-level model. Since only the crossbar and decoder are designed by MNSIM to support memritor-based computation, we use the parameter extracted from the layout of this part to validate the area model. As shown in Fig. 4, a 32 × 32 1T1R memristor crossbar and the proposed decoders are designed in 130nm CMOS technology. The area of the layout is 3420 um^2 (45um × 76um), while the estimation result is 2251 um^2. The large error rate of area estimation is mainly because that the layout design needs to remain some intervals, and the decoder circuit also needs to be aligned with the memristor cells. MNSIM introduces the validation result as a coefficient for area estimation, and users can provide the coefficient of their own technologies to obtain a more accurate estimation.

TABLE III
SIMULATION TIME OF SPICE AND MNSIM

Crossbar Size	16	32	64	128	256
SPICE(s)	5.35	13.76	41.62	169.12	678.2
MNSIM(s)	0.0007	0.0011	0.0030	0.0192	0.0348
Speed-Up	7642×	12509×	13873×	8088×	19489×

TABLE IV
THE DESIGN SPACE EXPLORATION OF A 2048 × 1024 NETWORK BASED ON DIFFERENT INTERCONNECT TECHNOLOGY, CROSSBAR SIZE, AND COMPUTATION PARALLELISM DEGREE

Optimization Target	Area	Energy	Latency	Computation Accuracy
Area(mm^2)	**12.183**	20.730	29.345	117.086
Energy per Operation(uJ)	35.90	**3.1923**	3.748	10.35
Latency(us)	43.43	0.5153	**0.3470**	10.35
Error Rate of Output(%)	17.98	17.98	17.98	**1.09**
Power(W)	0.8266	6.195	10.80	29.66
Crossbar Size	256	256	256	64
Line Tech Node	28	28	28	45
Parallelism Degree	1	128	256	64

B. Simulation Speed-up

We test the simulation time of single memristor crossbar with different sizes. As shown in Table III, MNSIM can get more than 7000× speed-up compared with SPICE. Moreover, the speed-up can further increase when simulating multiple crossbars and the large amount of peripheral circuits.

C. Case Study

We use a 2048 × 1024 network with 8-bit bipolar weights and 8-bit signals for case study. There are some researchers already shown that RRAM devices can implement 8-bit storage in single cell [22]. However, if the precision of devices can not directly support 8-bit storage, we can use two crossbars to store the highest 4-bit values and lowest 4-bit values separately and add the results by the adders in the peripheral module in the reference design of MNSIM. Considering that the precision is usually determined by the device, we do not use it as a parameter for optimization, and the experiments are based on 8-bit level RRAM model. This network scale is used in existing algorithm [13], which can support a wide design space of more than 10,000 designs. We use the reference design based on 45nm CMOS. The crossbar size, computation parallelism degree, and interconnect technology are three variables for design space exploration. Specifically, the computation parallelism degree means the amount of read circuits for each crossbar. For example, when the parallelism degree is 4, it means we simultaneously obtain the results of 4 columns for each crossbar. In this experiment, the crossbar size doubled increases from 4, 8, to 1024; the computation parallelism degree ranges from 1 to 128; and the interconnect technology(nm) is chosen from {18,22,28,36,45}. MNSIM uses traversal method for optimization due to the fast simulation speed. All the 10,220 designs can be simulated within 4 seconds in this case.

1) Design Space Exploration: Since the computing error rate of memristor crossbar is more than 2000% when the crossbar size is 1024 with 18nm interconnect lines, we set up that the computing error rate of memristor crossbar cannot be larger than 25% in the experiment. The design space exploration results are shown in Table V. Each column of the table shows an optimized design aimed at a specific optimization target. For example, the first column is the the performance factors and circuit details of the smallest area. Compared with 2nd and 3rd columns, the 1st column has less area and power consumption with the same interconnect technology and crossbar size. This is because it reads the computing results one by one, but the latency increases and the energy of whole computation grows back. From the 4th column, we can see that the most accurate computation is implemented by large interconnect technology and a middle crossbar size, which accords with our previous analysis. In addition, the result in Table V is not comparable with the performance of RRAM-based memory, because the computation-

TABLE V
THE TRADE-OFF BETWEEN AREA, POWER, AND ACCURACY BASED ON DIFFERENT CROSSBAR SIZES

Crossbar Size	256	128	**64**	32	16	8
Error Rate(%)	7.71	2.07	**1.09**	1.46	2.38	3.50
Area(mm^2)	29.34	58.59	117.11	234.10	468.32	936.81
Energy(uJ)	3.74	5.94	10.35	19.21	37.09	73.38

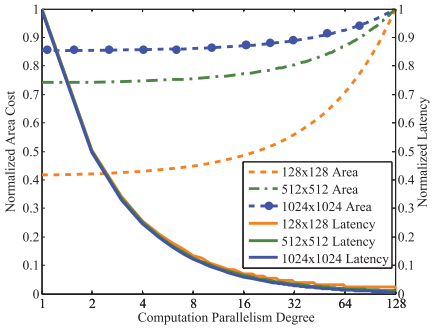


Fig. 5. The influence of computing parallelism degree on area and latency with different crossbar sizes. The area and latency results are normalized by the maximum value of each crossbar size for comparison.

oriented circuit is quite different with memory as discussed in Section II-B.

2) Trade-Off Among Area, Power, and Accuracy: To further analyze the trade-off results, we use the 45nm interconnect technology as an example to analyze the relationship between area, power, and computing accuracy influenced by crossbar size. Obviously, if we use larger crossbar size in each unit, the amount of units gets smaller. Each splitting of rows in weight matrix leads to an increase of peripheral circuits such as DACs and read circuits, so the power and area decrease when using larger crossbar. However, large crossbar suffers from the impact of non-ideal factors as discussed in Section IV-C, so there is a trade-off between computing accuracy and other performance. Table IV shows the results among different crossbar sizes. We can get computing accuracy improvement at the cost of area and power only when the crossbar size is larger than 64. When the crossbar is too small, the parallel resistance of a column grows up. As a result, the actual voltage between each cell gets smaller according to Eq. (6). The change of voltage leads to the change of resistance by the non-linear V-I characteristic of memristor, which influences the computing accuracy.

3) Trade-Off Between Latency and Area: There is also a trade-off between latency and area. The output peripheral circuits can be shared by multiple columns to reduce the area and power, but the latency increases because each unit needs to compute many times. This trade-off is also influenced by the *column* amount of crossbar. Fig. 5 shows the area and latency results when using different computation parallelism degrees and different crossbar sizes, where each line shows the results of the same crossbar size. Considering that the results of different crossbar sizes vary a lot in absolute number, we normalize them by the maximum value of each crossbar size's result. When the parallelism degree goes down, the increasing trend of latency is similar in different crossbar sizes but the area reduction trend varies. This is because the unit amount is small with large crossbar size, so the area of neurons and peripheral circuits takes large proportion of area, which limits the gain by reducing read circuits. The trade-off between area and latency is shown in Fig. 6. We can obtain large area reduction at the cost of little latency, and there is an inflection point for each crossbar size.

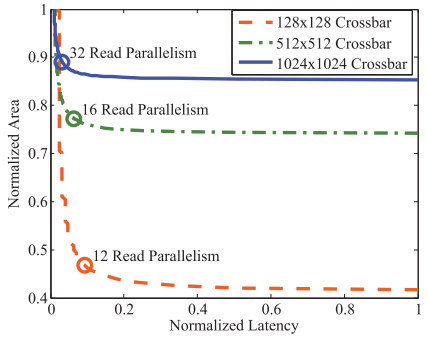


Fig. 6. The trade-off relationship between area and latency with different computation parallelism degrees and different crossbar sizes.

VI. CONCLUSIONS

In this work we propose MNSIM, the first behavior-level simulation platform for memristor-based neuronmorphic computing system. MNSIM proposes a scalable hierarchical structure of memristor-based neuromophic computing system, and provides flexible interface for users to customize their designs in multiple levels. MNSIM also provides a detailed reference design for large-scale applications. A behavior-level model is proposed to estimate the computing accuracy of memristor-based structure, and the error rate of this model is less than 1% compared with SPICE result. The experiment results show that MNSIM can reach more than 7000× speed-up with SPICE. MNSIM can also provide the trade-off between different designs and estimate the optimal performance. In the future, we will test MNSIM in larger networks, and further support RRAM-based structure designs for other applications like Spiking Neural Network [23] and Convolutional Neural Network.

REFERENCES

[1] Hadi Esmaeilzadeh et al. Neural acceleration for general-purpose approximate programs. In *MICRO 2012*, pages 449–460.
[2] Yuan Xie. Future memory and interconnect technologies. In *DATE 2013*, pages 964–969.
[3] Hadi Esmaeilzadeh et al. Dark silicon and the end of multicore scaling. In *ISCA 2011*, pages 365–376.
[4] Yue Bai et al. Low power w: Alox/wox bilayer resistive switching structure based on conductive filament formation and rupture mechanism. *Appl Phys Lett*, 102(17):173503, 2013.
[5] Sung Hyun Jo et al. Nanoscale memristor device as synapse in neuromorphic systems. *Nano letters*, 10(4):1297–1301, 2010.
[6] Boxun Li et al. Memristor-based approximated computation. In *ISLPED 2013*, pages 242–247.
[7] Nathan Binkert et al. The gem5 simulator. *ACM COMP AR*, 39(2):1–7, 2011.
[8] Xiangyu Dong et al. Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. *TCADICS*, 31(7):994–1007, 2012.
[9] Matthew Poremba et al. Nvmain: An architectural-level main memory simulator for emerging non-volatile memories. In *VLSI 2012*.
[10] Wei Fei et al. Design exploration of hybrid cmos and memristor circuit by new modified nodal analysis. *TVLSI*, 20(6):1012–1025, 2012.
[11] Boxun Li et al. Rram-based analog approximate computing. *TCAD*, 34(12):1905–1917, Dec 2015.
[12] Tianqi Tang et al. Spiking neural network with rram: can we use it for real-world application? In *DATE*, pages 860–865, 2015.
[13] Dan Claudiu Ciresan et al. Deep big simple neural nets excel on handwritten digit recognition. *arXiv preprint arXiv:1003.0358*, 2010.
[14] Peng Gu et al. Technological exploration of rram crossbar array for matrix-vector multiplication. In *ASP-DAC 2015*, pages 106–111.
[15] Ximeng Guan et al. A spice compact model of metal oxide resistive switching memory with variations. *IEEE ELECTR DEVICE L*, 33(10):1405–1407, 2012.
[16] Cagli Carlo et al. Evidence for threshold switching in the set process of nio-based rram and physical modeling for set, reset, retention and disturb prediction. In *IEDM 2008*, pages 1–4.
[17] Boxun Li et al. Merging the interface: Power, area and accuracy co-optimization for rram crossbar-based mixed-signal computing system. In *DAC 2015*.
[18] Y Sasago et al. Cross-point phase change memory with 4f 2 cell size driven by low-contact-resistivity poly-si diode. In *VLSIT 2009*, pages 24–25.
[19] Jing Li et al. A novel reconfigurable sensing scheme for variable level storage in phase change memory. In *IMW 2011*, pages 1–4.
[20] Brahmantyo Heruseto et al. Embedded analog cmos neural network inside high speed camera. In *ASQED 2009*, pages 144–147.
[21] Ye Zhang et al. Metallic to hopping conduction transition in ta2o5-x/taoy resistive switching device. *Appl Phys Lett*, 105(6):063508, 2014.
[22] Ligang Gao et al. A high resolution nonvolatile analog memory ionic devices. In *NVMW*, pages paper–57, 2013.
[23] Yu Wang et al. Energy efficient rram spiking neural network for real time classification. In *GLSVLSI*, pages 189–194, 2015.