RENO: A High-efficient Reconfigurable Neuromorphic Computing Accelerator Design*

Xiaoxiao Liu, Mengjie Mao, Beiye Liu, Hai Li, Yiran Chen
University of Pittsburgh
Pittsburgh, USA
{xil116, mem231, bel34, hal66, yic52}@pitt.edu

Boxun Li, Yu Wang
Tsinghua University
Beijing, P.R. China
{lbx13, yu-wang}@mails.tsinghua.edu.cn

Hao Jiang
San Francisco State University
San Francisco, USA
jianghao@sfsu.edu

Mark Barnell, Qing Wu
Air Force Research Laboratory
Rome, USA
{mark.barnell.1, qing.wu.2}@us.af.mil

Jianhua Yang
University of Massachusetts
Amherst, USA
jjyang@umass.edu

ABSTRACT

Neuromorphic computing is recently gaining significant attention as a promising candidate to conquer the well-known von Neumann bottleneck. In this work, we propose **RENO** – a efficient reconfigurable neuromorphic computing accelerator. RENO leverages the extremely efficient mixed-signal computation capability of memristor-based crossbar (MBC) arrays to speedup the executions of artificial neural networks (ANNs). The hierarchically arranged MBC arrays can be configured to a variety of ANN topologies through a mixed-signal interconnection network (M-Net). Simulation results on seven ANN applications show that compared to the baseline general-purpose processor, RENO can achieve on average 178.4× (27.06×) performance speedup and 184.2× (25.23×) energy savings in high-efficient multilayer perception (high-accurate auto-associative memory) implementation. Moreover, in the comparison to a pure digital neural processing unit (D-NPU) and a design with MBC arrays cooperating through a digital interconnection network, RENO still achieves the fastest execution time and the lowest energy consumption with similar computation accuracy.

1. INTRODUCTION

Traditional von Neumann computers require frequent data exchanging between processors and memory chips. This design severely limits the system performance and efficiency, especially in computation-intensive cognitive applications. As a promising candidate to overcome the inefficiency of von Neumann architecture, neuromorphic systems recently became a hot research area in future tera-scale computing. Many studies have been conducted on the hardware implementation of *artificial neural networks* (ANNs) across both digital and analog domains. Examples include neural network accelerators for signal processing [5], digital approximate computing accelerators that leverage neural network algorithms [8], and heterogeneous systems built with GPUs and APUs for deep learning accelerations [9]. However, traditional CMOS technology has been proven to be inefficient for neuromorphic system design as dozens of transistors are usually required to build one neuron [5].

Discovery of nanoscale memristor devices [6] inspired an exciting approach to implement neuromorphic systems. Particularly, the similarity between the programmable resistance state of memristors and the variable synaptic strengths of biological synapses dramatically simplify the circuit realization of neural network models. The specialty of memristors has been investigated and exploited in a few research works that focus on either the circuit implementation of the matrix-vector multiplications in conventional approximate computing acceleration [16, 17].

In this work, we propose ***RENO*** – a novel efficient reconfigurable neuromorphic computing accelerator. RENO uses on-chip memristor-based crossbar (MBC) arrays to implement a perceptron networks, aiming at the acceleration of ANN computations. Unlike many neuromorphic systems that perform the computations on pure digital ALUs or analog approximate computing units with AD/DA interface, our design adopts a hybrid method in data representation: the computation within the MBC arrays and the signal communications among the MBC arrays are conducted in analog form, while the control information remains as digital signals. Compared to the existing implementations of digital ANN accelerators and approximate computing units, the key distinctions of RENO can be summarized as:

- ***A efficient memristor-based mixed-signal accelerator*** is designed to speed up neuromorphic computing and support the implementations of a variety of neural network topologies;
- ***A mixed-signal interconnection network (M-Net)*** is proposed to assist the communication of computational signals among the MBCs.
- ***An optimized configuration*** is discussed and finalized by thoroughly analyzing the impact of various design parameters on the system performance/accuracy.

RENO offers a cost-efficient and fault-tolerant ANN computation platform complementing the general computations of CPU cores. In the evaluations of RENO, we adopt a set of prevailing ANN benchmarks and two ANN topologies-*Multilayer perception* (MLP) and *auto-associative memory* (AAM) to demonstrate the tradeoffs of the computation performance and accuracy for different RENO configurations. Simulation results show that compared to the baseline

*This work is supported in part by NSF XPS-1337198, NSF CNS-1116171, AFRL FA8750-15-2-0048, DARPA D13AP00042, HP Lab Innov. Res. Pgm, NSFC 61373026, and Tsinghua Univ. Init. Sci. Res. Pgm. Received and approved for public release by AFRL on 03/04/2015, case number 88ABW-2015-0833. Any Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or its contractors.

DAC '15, June 07 - 11 2015, San Francisco, CA, USA

ACM 978-1-4503-3520-1/15/06$15.00
http://dx.doi.org/10.1145/2744769.2744900.

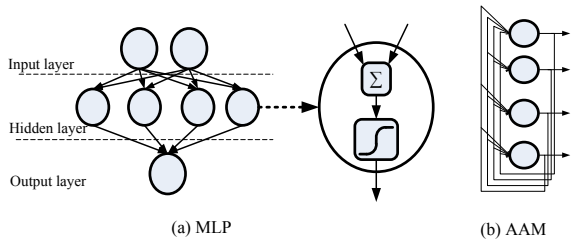

Figure 1: (a) A 3-layer MLP; (b) A 1-layer AAM with 4 neurons.

general-purpose CPU, RENO with MLP (AAM) configuration can achieve on average 178.4× (27.06×) performance speedup and 184.2× (25.23×) energy saving over the selected ANN applications. Furthermore, we compare RENO (MBC arrays with M-net) to a *digital neural processing unit* (D-NPU) [8] and a conventional MBC-based neuromorphic accelerator design, i.e., MBC arrays with *digital routing network* (D-Net). The results show that RENO achieves the fastest execution time and the lowest energy consumption while maintaining comparable computation accuracy.

2. PRELIMINARY

2.1 Artificial Neural Network (ANN)

In this work, we consider two typical ANNs – MLP with high efficiency and AAM with high accuracy. Both of them are simplified mathematical models of biological neural networks. MLP belongs to the feedforward ANNs that are widely utilized in approximate computing [11]. It maps a set of input data to outputs through multiple layers of nodes in a directed graph. Figure 1(a) shows an example of a three-layer MLP. The input nodes collect and convey the input bits to the following layer through the weighted connections. Except for the input nodes, each node represents a *neuron* with a nonlinear activation function, e.g., a sigmoid function $f(x) = \frac{1}{1+e^x}$ on the sum of all the signals it receives.

AAM is normally used as recurrent neural networks, performing pattern recognition and completion etc. [11]. Figure 1(b) shows a Hopfield network acting as an AAM. An input vector distorted by noises or other randomness will go through the network iteratively and converge to the closest version of the vector pattern. In general, the non-iterative MLP executes faster than the iterative AAM while the latter one is much more dependable due to its inherent fault tolerance characteristic.

As an important operation of ANN, *training* determines the weight associated with each connection and prepares the ANN to respond to certain unseen data with desired outputs. In this work, we adopt the back-propagation and the delta rule [16] to perform the training. Note that our works mainly focus on the testing/computation of the ANN by assuming RENO has been trained by supervised algorithms for specific applications.

2.2 Memristor and Memristor-based Crossbar

Memristor is regarded as the 4^{th} fundamental circuit element whose resistance (*memristance*) is determined by the total electric charge/flux through it [6]. In theory, a memristor can be programmed to any arbitrary resistance state within its lowest and highest bounds by appropriately controlling the amplitude and duration of the programming signal. Recent research has obtained 7-bit programming resolution on memristors [4] with sophisticated peripheral circuits.

Similar to biological synapses, a memristor device can

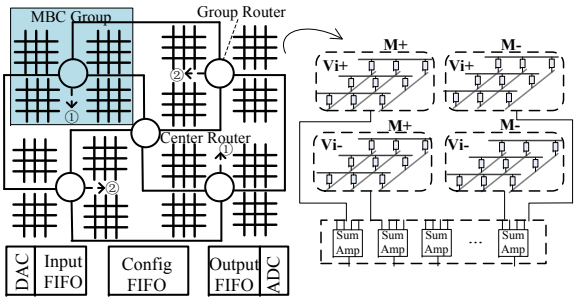

Figure 2: The RENO architecture.

"record" the historical profile of the applied excitations as its resistance changes. This feature inspired many studies on memristor-based synapse designs. For example, memristors can be employed in spiking networks and trained by using the *spike timing dependent plasticity* (STDP) learning rule [14]. The studies also presented the use of MBCs in percetpron network construction and demonstrated extremely efficient, accurate, and fast ANN implementations [12].

Figure 2(a) depicts the diagram of a MBC which represents the connections between two layers in a MLP. The input ($\mathbf{V}_i$) and output voltages ($\mathbf{V}_o$) follow the relation:

$$\mathbf{V}_o = \mathbf{C} \times \mathbf{V}_i. \quad (1)$$

Here $\mathbf{C}$ is the connection matrix. Due to device variations and physical constraints in real implementation, however, the relationship between the connection matrix and the resistance matrix of a MBC cannot achieve a perfect one-to-one mapping as Eq.(1) but rather obey an approximation. Implementing a N-layer MLP requires $N-1$ MBCs connected in series. A large volume of ANN computations (i.e., weight multiplications) can be simultaneously performed by the MBCs in analog form without any internal control logics.

In this work, we adopt the MBC programming method in [12] where an adaptive write driver [4] is used to program the memristors to particular resistance states. As memristors on the same row can be tuned simultaneously, the impact of sneak path leakage can be significantly suppressed.

3. THE RENO ARCHITECTURE

Figure 2 depicts the proposed RENO structure. It works as a complementary functional unit to CPU and particularly accelerates ANN-relevant executions. In this design, *memristor-based crossbar* (MBC) arrays are used to perform analog neuromorphic computations. A *mixed-signal interconnection network* (M-net) is developed to connect the MBC arrays and conduct the topological reconfiguration of RENO. To receive command and data and send results back to the processor in digital form, *input*, *output* and *configuration FIFOs* are located at the interface of RENO.

3.1 Hierarchical Structure of MBC Arrays

As shown in Figure 2, MBCs are arranged in a *centralized mesh* (CMesh) manner to minimize the cost of the interconnection network.The example in the figure includes four MBC groups, each of which is formed with four MBC arrays connected through a group router. An MBC array is partitioned into four sub-crossbars to implement the multiplication of the combination of the signed signals and the signed synaptic weights [12]. An optimal MBC design may contain 64 rows and 64 columns which offers a good compromise between performance and reliability. In fact, this array scale covers the majority of learning applications, 80% of

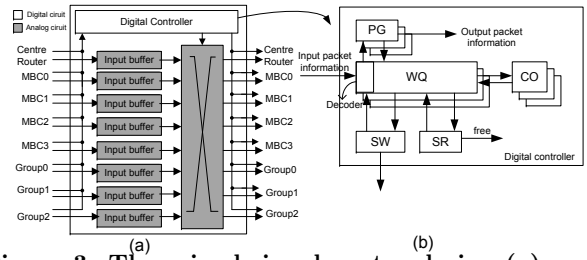

Figure 3: The mixed-signal router design (a) architecture (b) digital controller.

which have less than 60 neurons in the input layer [19]. Applications requiring larger connection matrices can be partitioned into smaller tasks and executed on multiple MBC arrays simultaneously or sequentially.

In this centralized hierarchical architecture, the data communication is performed at both inter-group and intra-group levels. The *central router* shown in Figure 2 connects to the CPU and all *group routers*. Each group router talks to the four local MBC arrays within the group, three other group routers, and the central router. This architecture offers easy routing paths between different components as well as a good design scalability.

3.2 Mixed-signal Interconnection Network

The signal transmission within RENO can be realized in either digital or analog form. Digital signal transfer has good controllability and supports high-frequency operations. However, as the computation of MBC arrays is in analog form, *digital-to-analog/analog-to-digital* (DA/AD) conversions are required at the interface of MBC arrays and routers, which inevitably degrades the signal precision and results in significant area and power overheads. The small footprint of the MBC arrays limits the data communication distance, e.g., within $0.53mm$ in our design, making it possible to transfer signals in also analog form. Moreover, the impact of signal distortion generated during the analog signal transmission on computation reliability can be tolerated by the intrinsic high fault resistance of ANN algorithms.

We propose a mixed-signal interconnection network called *M-Net* to assist the task mapping and data migration over the MBC arrays. M-Net maintains the data in analog form while it transfers the control and routing information in digital form so as to simplify the synchronization and communication between CPU and RENO.

Router Design. Figure 3(a) shows the group router design. Its analog data path consists of input buffers and data multiplexer/switches. Each input port can receive up to 64

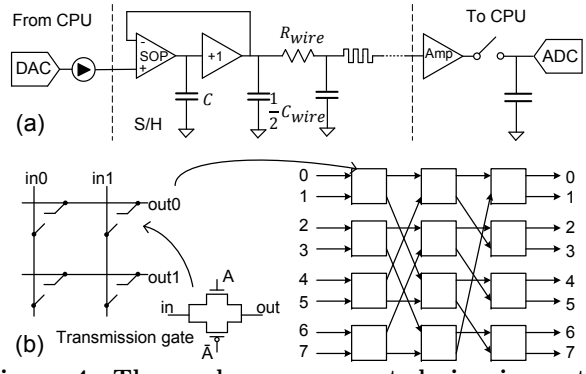

Figure 4: The analog component design in router: (a) the transmission path; (b) the multiplexer.

MLP	V	H	$Addr_0$[4:0]	$Addr_1$[4:0]	...	CPU Addr
AAM	V	H	$Addr_0$[4:0]	Loop[6:0]		CPU Addr

Figure 5: Routing information format.

analog signals corresponding to a set of the inputs/outputs of a MBC array, referred as *a packet*. During operations, a *switched-op-amp* (SOP) based *sample-and-hold* (S/H) circuit in Figure 4(a) [7] serves as an analog buffer, which holds and passes the analog data to the next destined MBC array or router. The S/H circuit adopts a pseudo-differential topology and turns off the transistor in the saturation region to minimizes the nonlinear distortion.

Figure 4(b) depicts the conceptual implementation of an 8×8 multiplexer using transmission-gate based analog crossbar switches. The multiplexer can dynamically establish the routing path from any input port to any output port under the guidance of the digital control logic. 64 copies of such a multiplexer are required at each port of a group router to transmit a packet of up to 64 signals simultaneously.

The digital controller of a router is shown in Figure 3(b). The routers in RENO are responsible for not only transferring data as a traditional *Network-on-Chip* (NoC) does, but also processing routing information. Thus, a *work queue* (WQ) is introduced. Once the WQ receives the routing information of a data packet, it will decode the information to generate the control signals for the other components in the router. The routing path configuration in the multiplexer is controlled through a *switch allocator* (SA). Each WQ entry is associated with a multi-bit *computing counter* (CO) to monitor the computation status of a local MBC array by counting the number of the executed loops. In this work, we utilize a 7-bit CO (supporting up to 128 loops) because all the selected benchmarks can complete executions within 100 loops. As a local MBC array approaches to the end of its computation, the CO notifies its WQ. The computation result will be sent to the CPU or another MBC group with the routing information generated by *packet generator* (PG). At this time, the corresponding WQ entry is released and the updated routing information remains in the group router. *Status recorder* (SR) logs and broadcasts the availability of a local MBC array to all the connected routers.

Routing Management. Figure 5 presents the format of the routing information adopted in RENO, including 1-bit valid bit (V), 1-bit routing field (H), address field ($Addr_i$), and looping field ($Loop$). An address field contains 5 bits: $Addr_i$[1:0] identifies the group router, $Addr_i$[3:2] denotes a MBC array within the group, and $Addr_i$[4] indicates if the data shall be sent back to CPU. According to the CO design, the looping field contains 7 bits supporting up to 128 loops.

Bit H represents the type of ANN implementations (MLP or AAM) and determines the format of routing information. The MLP configuration does not require a looping field. The address fields of MLP includes the addresses of the MBC arrays that the data will go through and the address representing the CPU. The AAM configuration needs both address and looping fields to guide the destined router address and the related number of computation loops, respectively. A routing information always ends at *CPU address*, indicating the completion of data transmission. Once an input data packet goes through the corresponding router, these address and looping fields can be recycled by PG.

4. EXPERIMENTAL METHODOLOGY

Table 1: The component parameters of RENO

Memristor					
R_L=200Ω, R_H=160KΩ, V_{th}=2V					
MBC Array & M-Net					
	Neuron logic	Network	MBC	DAC	ADC
Power	126μW	0.88μW	0.72μW	5.2mW	3.8mW
Speed	0.93ns	4.9ns	3.1ns	333MHz	333MHz

Area Estimation						
	RENO area	NoC (mm^2)			DAC/ADC	MBC
	(mm^2)	Input/output	Channel	Control	(mm^2)	(mm^2)
M-Net	0.943	0.598	0.014	0.252	0.072	0.007
D-Net	1.793	0.268	0.065	0.301	1.152	0.007

Table 2: The simulation platforms

RENO	***CU***	4 MBC groups, 4 MBC arrays/group, 4 MBC/array, MBC size: 64×64
	IO	64×4-bit In-queue/Out-queue, 128×64-bit Config-queue, 64 parallel DACs, 64 parallel ADCs
	M-Net	Mixed-signal CMesh, 333 MHz for digital control
D-NPU	***CU***	16 digital PEs, Input/Output Buffer 64×4-bit, 4-bit Multiply-add unit, Weight Cache 4096×4-bit, Sigmoid Unit LUT 512×4-bit
	IO	64×4-bit In/Out-queue, 128×64-bit Config-queue
	D-Net	Digital CMesh, 1.332 MHz, 64-bit datapath

4.1 Circuit Implementation & Simulation

We created a Verilog-A memristor model by adopting the device parameters from [14] and scaling them to a $65nm$ node based on the relation of device resistance and area. To achieve high speed and small form factor, we adopt the flash *analog-digital converter* (ADC) and current steering *digital-analog converter* (DAC) [10] designs. The resolution is set to 4-bit to comply with the data resolution required by the selected benchmarks. We estimate the delay and power of all these components and extract the layout areas under Cadence Virtuoso environment. The detailed design parameters and area estimation can be found in Table 1.

The area of RENO is mainly occupied by the routers. The analog signal transmission in the concerned distance is simulated, e.g., among the routers. Our simulation shows that a voltage swing between $0V$ and $1V$ can be transferred from one end of an $0.53mm$ interconnect to the other end in $0.5ns$ by considering signal fluctuations and distorations. All the major noise resources, including $1/f$ noise in amplifier, thermal noise produced by memristor and amplifier, and quantization noise caused by ADC, have been evaluated. The result shows that the quantization noise up to $18mV$ dominates the overall noise. Such noise magnitude is much smaller than the resolution of 4-bit DAC/ADC ($62.5mV$) so that the introduced impact remains under a tolerable level. The device mismatch can be calibrated by a predetermined look-up table. Finally, an inline calibration scheme is designed to ensure run-time execution accuracy of RENO by monitoring the resistance shifting of MBC arrays during operation and restoring the resistance with a set of training vectors. The inline calibration can be conducted anytime between executions of two RENO inputs, and hence, does not affect the execution continuity of RENO and generates very marginal impacts on the execution time (e.g., <0.35% in MLP and <0.67% in AAM, respectively).

Reliability analysis is conducted using Monte-Carlo simulations. We assume both the resistance of the memristors and the analog inputs of the MBCs follow normal distributions. In each Monte-Carlo simulation, the initial memristor resistance of a MBC sample is fixed as it is decided by the offline training. However, the signal fluctuation is generated on-the-fly during the entire execution.

Table 3: The description and implementation details of the seven selected benchmarks

Benchmark	Training error		MBC usage		Topology
	MLP	AAM	MLP	AAM	MLP
cancer	0.02%	0.07%	2 in 1	2 in 1	36→16→2
connect-4	0.02%	0.08%	2 in 1	3 in 1	42→30→3
gene	0.09%	0.03%	6 in 2	12 in 3	120→100→3
lymphography	0.05%	0.02%	2 in 1	4 in 1	29→19→4
MNIST	0.35%	0.02%	5 in 2	10 in 3	64→128→32→10
mushroom	0.01%	0.01%	3 in 1	8 in 2	125→32→2
thyroid	0.15%	0.11%	2 in 1	3 in 1	21→32→3

4.2 Architecture Level Simulation Setup

We modify *MacSim* [1], a PIN-based cycle-level X86 simulator, by adding a cycle-accurate RENO module to conduct architecture level evaluations. The CPU is configured as an Intel Atom-like processor. A compilation flow similar to [8] is utilized to generate a RENO-aware binary of which the target ANN code is executed in RENO. Since all the selected benchmarks are ANN oriented, on average, 99% of execution time is consumed on running the target codes. Thus, in the following evaluations, the execution time of the target codes is used to represent the overall performance. Table 2 summarizes the parameters of our simulation platform. The energy consumption of the CPU core is estimated using *McPAT* [15]. We generate a detailed log of RENO utilization during the execution to calculate the energy consumption of RENO based on the circuit level simulation results. The data traffic and the power consumption of the M-Net are simulated by a modified *booksim* simulator [13].

We choose seven representative learning benchmarks described in Table 3. `Cancer`, `gene`, `mushroom` and `thyroid` are selected from Proben1 [18]. `connect-4` and `lymphography` from UCI machine learning repository [3] are tailored for neural network implementations. `MNIST` [2] is a widely used benchmark of learning and recognition algorithms[1]. We implement all the benchmarks within MLP and AAM models and measure the execution quality in classification rate. We define training error as the *mean square error* (MSE) between the actual and target outputs under the training vectors. Table 3 summarizes the implementation details and the initial training errors.

5. EXPERIMENTAL RESULTS

We investigate the design and optimization of RENO by thoroughly evaluating the impacts of MBC sizes, training effort, device variations and signal fluctuations. The quality of RENO from the perspectives of computation accuracy, performance, and energy consumption are also explored by comparing with general-purpose CPU and other two ANN accelerators: D-NPU and MBCs+D-Net.

5.1 MBC Training Effort

As the DAC/ADC resolution is capped by the resolution of computation data provided in the benchmarks, the computation accuracy and energy consumption of RENO are greatly influenced by the training effort, which can be measured by the size of a used training data set. For a specific benchmark, the computation accuracy can be improved by increasing the size of training data set. However, it even-

[1]The image of MNIST is compressed from 28×28 pixels into 8×8 pixels and the gray scale is reduced from 256 to 16.

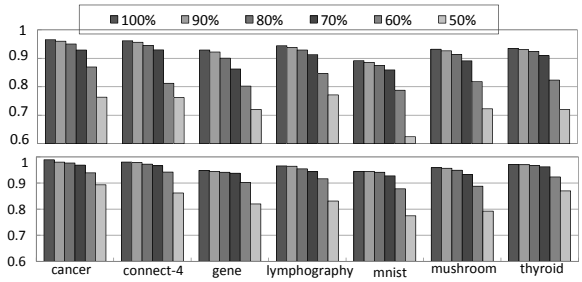

Figure 6: The normalized classification rates of (a) MLP and (b) AAM under different MBC training efforts. The DAC/ADC resolution is set to 4-bit.

tually gets saturated when the number of the training data reaches a threshold, i.e., the saturated training data set size.

Figure 6 (a) and (b) respectively compare the computation accuracy (i.e., the classification rate) degradations of MLP and AAM implementations under different training efforts. The classification rates have been normalized to the ideal baseline where the execution is performed by the floating-point unit of the CPU. Here the training effort is normalized to the saturated training data set size of each benchmark.

Generally, the MLP implementation is more sensitive to the reduction of training effort. In particular, `gene`, `mnist` and `mushroom` experience considerable reduction in classification rate due to their large network scale. Benefiting from the iterative feedback loop, the AAM implementation demonstrates much better computation accuracy than MLP under the same training effort. For AAM, the average classification rate of all benchmarks keeps above 83% even the training effort is as low as 50%. In the following evaluations, we set a training effort of 70% which can simultaneously satisfy the computation accuracy requirements of both MLP and AAM implementations and maintain reasonable hardware and performance overheads of training.

5.2 Device Variations & Signal Fluctuations

Figure 7 shows the impacts of device variations and signal fluctuations on the computation accuracy of RENO. Here σ_p denotes the standard deviation of memristor resistance incurred by process variations; σ_f denotes the standard deviation of the magnitude of the analog signals generated from DA/AD conversion, routing/buffering, sum-amplifier and sigmoid function. Since σ_f has greater impact on the

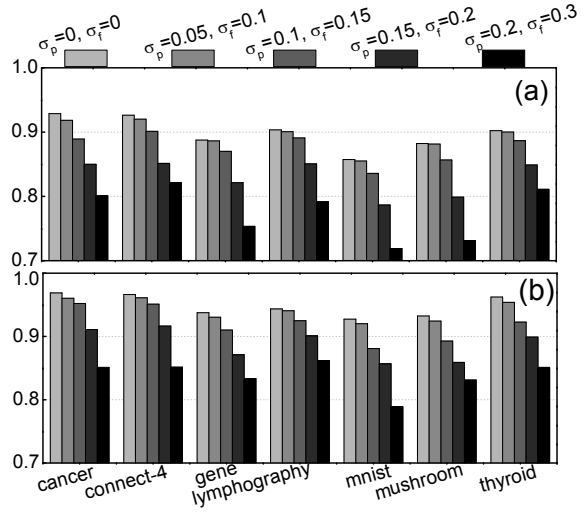

Figure 7: The impact of device variations and signal fluctuations on accuracy: (a) MLP, (b) AAM.

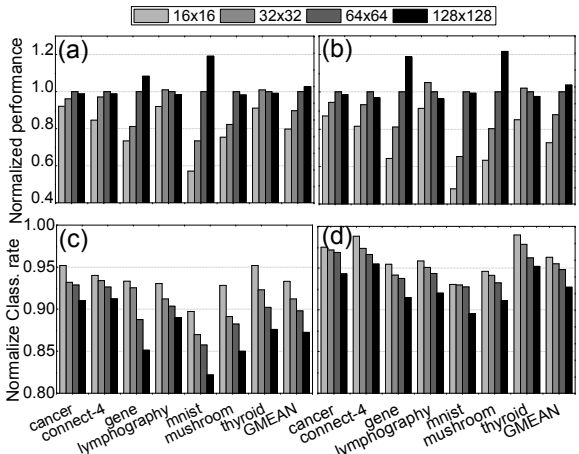

Figure 8: The normalized RENO performance and the classification rate (σ_p=0.05 and σ_f=0.1) at different MBC sizes in MLP (a,c) and AAM (b,d) configurations.

computation accuracy than σ_p [16], we choose very pessimistic settings of σ_f to cover even the extreme cases.

As expected, the increase of device variations and signal fluctuations generally degrades the computation accuracy of RENO with both MLP and AAM configurations. Interestingly, the normalized classification rate of `mnist` degrades slightly faster than other benchmarks, indicating a less robust ANN realization. Nonetheless, both MLP and AAM maintain a very moderate computation accuracy deterioration when σ_p and σ_f are within a realistic range, i.e., σ_p=0.05 and σ_f=0.1. Again, the AAM configuration demonstrates better tolerance to process variations and signal fluctuations than the MLP.

5.3 Impact of MBC Sizes

On one hand, increasing MBC size improves the computation efficiency of RENO as more calculations can be done simultaneously. It will also reduce the overheads of the computation partitioning and the signal routing among MBCs if the ANN size is larger than the MBC size. On the other hand, a larger MBC is more vulnerable to process variations and signal fluctuations.Moreover, when the MBC size exceeds the ANN size, part of power consumption and computation capacity of RENO will be wasted.

In Figure 8, we compare the execution time and the classification rate of all benchmarks when MBC size varies from 16×16 to 128×128. As the MBC size increases, the NCA performance of a particular benchmark keeps improving until the MBC size exceeds the largest scale of the ANN topology. Continuing increasing the MBC size does not further enhance the system performance. Nonetheless, the aggravated vulnerability of RENO to process variations and signal fluctuations at a large MBC size causes slight degradation on the classification rate, as shown in Figure 8(c,d). In this work, we selected 64×64 MBCs as the optimized configuration offering balanced computation efficiency and accuracy.

5.4 Comparison to Other Design Alternatives

We also explore the potential of RENO by comparing with other ANN accelerators. First, we construct a *digital neural processing unit* (D-NPU) which adopts the RENO topology but replaces MBC arrays and M-net with digital *processing elements* (PEs) [8] and a digital interconnection network *D-Net*, as shown in Figure 10. To perform a fair comparison,

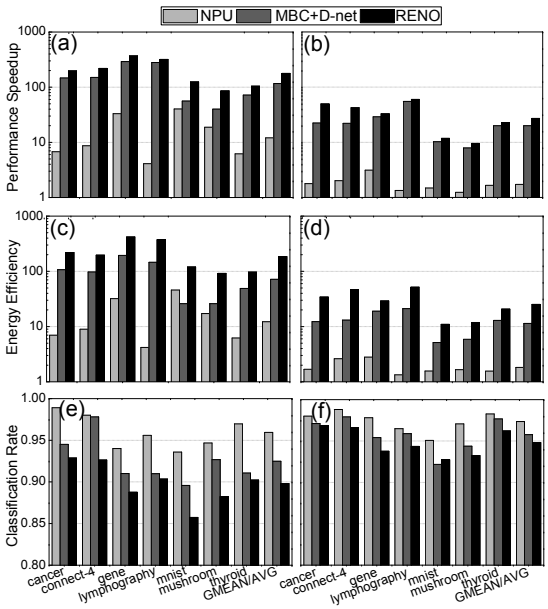

Figure 9: Comparisons of three ANN accelerators with MLP (a,c,e) and AAM (b,d,f) configurations.

the input/output FIFO and weight cache of each PE are scaled up to match the computational capacity of a MBC array. The PE latency and power are extracted from a Verilog-HDL model synthesized with 65nm library using VCS and Design Compiler. The detailed D-NPU configuration is depicted in Table 2. In addition, to study the efficacy of M-Net, we also construct an alternative design by solely replacing the M-Net in RENO with D-Net. The configuration of D-Net is estimated in *booksim* to offer the similar transmission capacity as M-Net. Meanwhile, DAC/ADC pairs are required at the interface of each router for the frequent DA/AD conversions before/after any MBC-based computation. Figure 9 compares the performance, energy efficiency, and classification rate of three ANN accelerator designs: D-NPU, MBC+D-Net, and RENO. Here, the energy efficiency is defined as the inverse of system energy consumption. The performance and energy efficiency are normalized to those obtained from the baseline CPU execution.

As shown in Figure 9 (a,b), RENO demonstrates the highest speedup due to the higher computation ability than D-NPU and the reduction of DA/AD conversions compared with MBCs+D-Nets: The *geometric mean speedup* (GMS) values rise to 178.41× (MLP) and 27.06× (AAM). Relatively speaking, the AAM configurations obtain less speedup due to the costly iterative computation. Figure 9(c,d) compare the energy efficiency of all the designs, which demonstrates a similar trend as the performance results. MLP and AMM configurations of RENO achieve on average 184.24× and 25.23× energy savings, respectively. RENO generally achieves more than 2× higher energy efficiency than that of

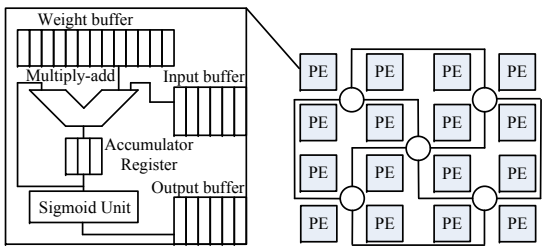

Figure 10: A D-NPU built with digital PEs in [8].

MBC+D-Net due to the dramatically reduced DA/AD conversion overhead. Figure 9(e,f) compare the classification rate of all the designs. In MLP results, RENO demonstrates the lowest computation accuracy as other designs benefit from digital data transferring and computation with better resolution. In AAM results, however, RENO obtains very high (i.e., 92%) average classification rates that are close to (i.e. discrepancy < 2.8%) that of D-NPU and MBCs with D-Net by paying the cost of more computing iterations.

6. CONCLUSION

In this work, we propose a reconfigurable memristor-based neuromorphic computing accelerator (RENO). Compared to a conventional CPU, RENO achieves on average 177.67× (27.2×) performance speedup and 184.71× (25.18×) energy reduction over the simulated benchmarks processed by MLP (AAM) neural networks. The computation accuracy degradation is well constrained within an reasonably low range. Although only two ANN configurations are presented, RENO can support a variety of ANNs by properly reconfiguring the M-Net and guiding data routing among the MBC arrays.

7. REFERENCES

[1] "Macsim," http://code.google.com/p/macsim/.
[2] "The mnist database," http://yann.lecun.com/exdb/mnist/.
[3] "Uci machine learning," http://archive.ics.uci.edu/ml/.
[4] F. Alibart *et al.*, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," *Nanotechnology*, vol. 23, no. 7, 2012.
[5] B. Belhadj *et al.*, "Continuous real-world inputs can open up alternative accelerator designs," in *ISCA*, 2013, pp. 1–12.
[6] L. O. Chua, "Memristor-the missing circuit element," *Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.
[7] L. Dai and R. Harjani, "Cmos switched-op-amp-based sample-and-hold circuit," in *IEEE Transactions on Solid-state circuits*, 2000.
[8] H. Esmaeilzadeh *et al.*, "Neural acceleration for general-purpose approximate programs," in *MICRO*, 2012, pp. 449–460.
[9] J. Gu *et al.*, "Implementation and evaluation of deep neural networks (dnn) on mainstream heterogeneous systems," in *APSys*, 2014, p. 12.
[10] M. Gustavsson, J. J. Wikner, and N. Tan, *CMOS data converters for communications*, 2000.
[11] S. O. Haykin, *Neural Networks and Learning Machines*. London: Prentice Hall, 2008.
[12] M. Hu *et al.*, "Hardware realization of bsb recall function using memristor crossbar arrays," in *DAC*, 2012, pp. 498–503.
[13] N. Jian *et al.*, "A detailed and flexible cycle-accurate network-on-chip simulator," in *ISPASS*, 2013, pp. 86–96.
[14] 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.
[15] S. Li *et al.*, "Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *MICRO*, 2009, pp. 469–480.
[16] B. Liu *et al.*, "Digital assisted noise eliminating training for memristor crossbar based analog neuromorphic computing engine," in *DAC*, 2013, pp. 1–6.
[17] X. Liu *et al.*, "A heterogeneous computing system with memristor-based neuromorphic accelerators," in *HPEC*, 2014, pp. 1–6.
[18] L. Prechelt, "Proben1-a set of neural network benchmark problems and benchmarking rules," University of Karlsruhe, Tech. Rep., 1994.
[19] O. Temam, "A defect-tolerant accelerator for emerging high-performance applications," in *ISCA*, 2012, pp. 356–367.