XNOR-RRAM: A Scalable and Parallel Resistive Synaptic Architecture for Binary Neural Networks

Xiaoyu Sun, Shihui Yin, Xiaochen Peng, Rui Liu, Jae-sun Seo, and Shimeng Yu

School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85281, USA
Email: jaesunseo@asu.edu, shimengy@asu.edu

***Abstract* – Recent advances in deep learning have shown that Binary Neural Networks (BNNs) are capable of providing a satisfying accuracy on various image datasets with significant reduction in computation and memory cost. With both weights and activations binarized to +1 or -1 in BNNs, the high-precision multiply-and-accumulate (MAC) operations can be replaced by XNOR and bit-counting operations. In this work, we propose a RRAM synaptic architecture (XNOR-RRAM) with a bit-cell design of complementary word lines that implements equivalent XNOR and bit-counting operation in a parallel fashion. For large-scale matrices in fully connected layers or when the convolution kernels are unrolled in multiple channels, the array partition is necessary. Multi-level sense amplifiers (MLSAs) are employed as the intermediate interface for accumulating partial weighted sum. However, a low bit-level MLSA and intrinsic offset of MLSA may degrade the classification accuracy. We investigate the impact of sensing offsets on classification accuracy and analyze various design options with different sub-array sizes and sensing bit-levels. Experimental results with RRAM models and 65nm CMOS PDK show that the system with 128×128 sub-array size and 3-bit MLSA can achieve accuracies of 98.43% for MLP on MNIST and 86.08% for CNN on CIFAR-10, showing 0.34% and 2.39% degradation respectively compared to the accuracies of ideal BNN algorithms. The projected energy-efficiency of XNOR-RRAM is 141.18 TOPS/W, showing ~33X improvement compared to the conventional RRAM synaptic architecture with sequential row-by-row read-out.**

I. INTRODUCTION

Deep neural networks (DNNs) have shown remarkable performance in various intelligent applications including computer vision and speech recognition. However, the high demands on memory storage capacity and computational power make it unsuitable to implement the state-of-the-art DNNs on resource-limited devices such as embedded systems and mobile devices. For example, ResNet-50 [1] has 25.5M parameters and requires 3.9G high precision operations to classify one image and these numbers are higher for even deeper networks. Thus, it is prohibitive to directly implement the entire DNN on-chip, and the intensive data movements between on-chip processor and off-chip memory (e.g., DRAM) access becomes the bottleneck of the system performance and energy efficiency. Recently, deep learning researchers have demonstrated that Binary Neural Networks (BNNs) [2-3] are able to achieve satisfying classification accuracy on representative image datasets (e.g., MNIST, CIFAR-10, and ImageNet). The memory storage of these BNNs is significantly reduced since both the weights and neuron activations are binarized to -1 or +1, as compared to floating-/fixed-point precision. Furthermore, high-precision multiply-and-accumulate (MAC) operations can be replaced by bit-wise XNOR and bit-counting operations, drastically reducing the computational resources as well. Therefore, BNNs provide a promising solution for on-chip implementation of DNNs.

In hardware accelerators of DNNs, SRAM is commonly utilized to store synaptic weights in CMOS ASIC designs [4]. Nevertheless, a SRAM cell consumes more than 200 F^2 (F = technology feature size) in area, leading to a limited capacity of on-chip weight storage. To enhance on-chip storage, researchers have proposed using embedded non-volatile memories (eNVMs) with much less area (<$10F^2$) such as resistive random access memory (RRAM) [5] and phase change memory (PCM) [6] to implement "analog" synaptic weights. Despite holding great advantages on area-efficiency and static power reduction compared to SRAM, the non-ideal analog weight characteristics (e.g., weight update nonlinearity, limited dynamic range and precision) introduce significant accuracy degradation [6-7]. Hence, it is more practical to use technologically more mature binary eNVMs that have been demonstrated at Gb chip-level by industry as a near-term solution [8]. The prior work in [9] experimentally demonstrated BNNs (a two-layer perceptron) on a 16Mb RRAM macro chip with row-by-row sequential read-out of binary RRAM cells, showing ~96.5% accuracy on MNIST dataset. To get rid of the row-by-row sequential read-out, the pseudo-crossbar array with 1-transistor-1-resistor (1T1R) or the true crossbar array could allow fully-parallel read-out by activating all the word lines (WLs) simultaneously for the weighted sum (or matrix-vector multiplication) operation [10-11].

Theoretically, the binary activation in BNNs could allow using 1-bit sense amplifiers (SAs) instead of analog-to-digital converters (ADCs) to serve as the binary neuron. However, due to the intrinsic offset of the SAs introduced by process variation, the sensing failure becomes intensified when the column current increases [12] when multiple WLs are activated in the parallel read-out scheme. This may substantially degrade the classification accuracy as the threshold of binary neuron may differ from the ideal value in algorithms, leading to a constraint on the column length or the array size. To overcome this design challenge, array partition has to be adopted to split a large-scale matrix into multiple small sub-arrays. In this way,

978-3-9819263-0-9/DATE18/©2018 EDAA

ADC-like multi-level sense amplifiers (MLSAs) are exploited to generate the partial sums of sub-arrays, which are eventually added up to be the final sum for binary activation. The previous works in [13-14] presented RRAM based implementation of BNNs, however to the best of our knowledge, the practical design issues such as the impact of sensing offset and design tradeoffs related to the size of sub-array and bit-level of MLSA have not been discussed in those works. In this paper, we propose a parallel RRAM synaptic architecture with a bit-cell design of complementary word lines that implements XNOR and bit-counting operations. We benchmark the performance of the proposed architecture and analyze the tradeoffs between different design options. The contribution of this work includes:

- A parallel architecture (XNOR-RRAM) that integrates RRAM synaptic array and MLSA with optimized array partition for implementing deep BNNs of arbitrary size.
- Analysis of different sizes of RRAM sub-array and different bit-levels of MLSA by Monte Carlo simulations using TSMC 65nm CMOS PDK.
- With balanced tradeoffs, the optimized system with 128×128 sub-array size and 3-bit MLSA can achieve accuracies of 98.43% for MLP on MNIST and 86.08% for CNN on CIFAR-10, showing 0.34% and 2.39% degradation respectively compared to the accuracies of ideal BNN algorithms.
- Compared to a baseline design with sequential row-by-row read-out, the proposed parallel XNOR-RRAM architecture achieves 141.18 TOPS/W energy efficiency with an improvement factor of ~33X.

The rest of this paper is organized as follows: Section II proposes the XNOR-RRAM bit-cell array and architecture design, and discusses the practical design issues including the impact of SA offset on accuracy, the size of sub-array, and the bit-level of MLSA. Section III presents the benchmark results of the classification accuracy on MNIST and CIFAR-10 datasets considering the impact of hardware constraints and non-idealities. Section IV compares the performance on area, latency, and energy between sequential and parallel XNOR-RRAM architectures. Section V concludes the paper.

II. Parallel XNOR-RRAM Architecture Design

A. Binary Neural Network (BNN)

In a BNN, both the weights and neuron activations are binarized to -1 or +1. Therefore, multiplications between activations and weights can be simplified as XNOR operations and accumulation of the products is equivalent to bit-counting operation. In this paper, we trained BNNs using the algorithm proposed in [2] on the Theano platform. A multilayer perceptron (MLP) with a structure of 784-512-512-512-10 and a convolutional neural network (CNN) with 6 convolution layers and 3 fully-connected layers are trained for evaluations on MNIST and CIFAR-10 datasets, respectively. Table I presents the corresponding classification accuracy with floating point (FL) precision and binary precision for these two networks. For MLP on MNIST, the accuracy slightly drops from 99.00% to 98.77%; for CNN on CIFAR-10, the accuracy slightly decreases from 89.98% to 88.47%. Such minor degradations have also been observed in state-of-the-art BNN algorithms [2-3].

B. RRAM Based Synaptic Array

Fig. 1 shows the proposed bit-cell design for XNOR-RRAM implementation in (a), the diagram of sequential RRAM and parallel XNOR-RRAM architectures in (b) and (c), respectively. In this work, we only consider the pseudo-crossbar with 1T1R array since the two-terminal threshold switch selectors for true crossbar array are currently not technologically mature for large-scale integration. After offline training, one-time write operation is performed to load all the binary weights to the array for inference. The convolution and the vector-matrix multiplication in fully connected layers are

TABLE I. CLASSIFICATION ACCURACY IN DIFFERNET CASES

Network	Dataset	FL Precision	Binary Precision
MLP	MNIST	99.00%	98.77%
CNN	CIFAR-10	89.98%	88.47%

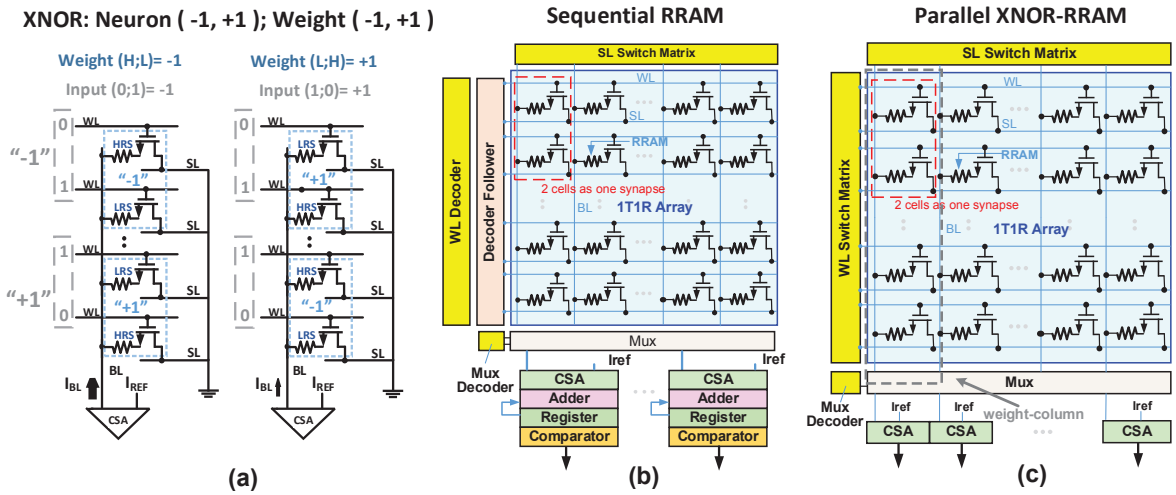

Fig. 1. (a) The customized bit-cell design for XNOR implementation. (b) The diagram of conventional sequential RRAM synaptic architecture. (c) The diagram of proposed parallel XNOR-RRAM architecture.

essentially the same operations if a 2D convolution kernel is unrolled into a 1D column [15].

Fig. 1(a) presents the principle of the proposed bit-cell design for XNOR-RRAM implementation. For each synaptic weight, -1 is represented by two cells where the top one is in HRS and the bottom one is in LRS. The reversed pattern is used for +1. For the WL input pattern, two adjacent WLs for each weight-cell are in complimentary state where (0, 1) represents -1 and (1, 0) represents +1. In this way, the value of the current that flows through each weight-cell during read-out is dependent on the combination of WL input pattern and bit-cell pattern. For example, when input vector is -1, for the cell of weight -1, the cell in the activated row is in LRS, leading to a large cell current, which can be regarded as a bit-wise XNOR output of "+1". For the cell of weight +1, the cell in the activated row is in HRS, leading to a small cell current, which can be regarded as a bit-wise XNOR output of "-1". When multiple WLs are activated in parallel, the LRS-cells will dominate the total bit line current (I_{BL}) if the on/off ratio of RRAM is sufficiently large. Consequently, I_{BL} will be proportional to the bit-counting results equivalent to the number of LRS-cells along the column. For example, 50% "+1" and 50% "-1" will lead to a final weighted sum of 0. Assuming the column length of the sub-array is 64, the sum of 0 can be mapped to the I_{BL} = 32 activated LRS-cells. Therefore, the reference current (I_{REF}) for the current sense amplifier (CSA) could be set to 32 LRS-cells' current for the binary neuron activation. If I_{BL} is smaller than I_{REF} that generates a CSA output "-1", it represents that there are more "-1" than "+1" along the column, and vice versa.

For the sequential RRAM design in Fig. 1(b), the encoded input neuron vector is fed into WL decoder, and only one WL is activated in each read-out operation. During the read-out operation, V_{BL} is biased to be ground, CSA imposes current on the selected bit line (BL) and compares I_{BL} with the fixed I_{REF} to determine the output. For each weight column, there are MAC units such as adder and register pair at the end of the column for row-by-row summation and partial weighted sum storage. In the end, the final weighted sum goes through a digital comparator to generate 1-bit neuron output (+1 or -1). For the parallel XNOR-RRAM design in Fig. 1(c), instead of a normal decoder, a WL switch matrix is designed to activate multiple WLs simultaneously depending on the input vectors to enable the parallel read-out operation. The parallel XNOR-RRAM architecture leverages the analog current summation to effectively realize the MAC operation, thus the adder/register periphery of the sequential row-by-row scheme is eliminated.

C. Impact of CSA Offset on Classification Accuracy

Theoretically, a 1-bit CSA can serve as the binary neuron for each column in parallel XNOR-RRAM to generate the binary neuron output. However, due to the intrinsic offset of CSA, the sensing pass rate (percentage of accurate sensing results) becomes worse when I_{BL} increases (as cell currents are summed up for a large array), which may bring significant accuracy degradation as the threshold of the neuron may differ from the ideal value in algorithms. In this section, we perform Monte Carlo (MC) simulations using TSMC 65nm CMOS PDK to investigate the impact of 1-bit CSA offset on classification accuracy of the MLP, using a 512×512 RRAM array. A current-latch based CSA [12] is employed, comprising precharge PMOS, cross-coupled pair, and pull-down NMOS as shown in Fig. 2(a). During the precharge phase, CSA imposes large precharge current to raise the voltage on BL/BL_B to drive I_{BL} and I_{REF}. When the difference between I_{BL} and I_{REF} reaches its maximum, the precharge transistors turn off and the cross-coupled pair compares the current difference to determine the output value. The offset of CSA is mainly due to the trip-point voltage mismatch between P1-N1 and P2-N2 that is caused by process variation. In the simulation setup, LRS/HRS resistance is assumed to be 200KΩ/200MΩ. The waveform in Fig. 2(b) shows the case of sensing 40 LRS-cells (BL) against 32 LRS-cells (BL_B) as an example. As I_{BL} is larger than I_{REF}, the voltage at node Q_B drops to the trip-point voltage earlier than node Q, raising node Q toward VDD. As a result, D_{OUT} remains at VDD while D_{OUT_B} drops to "0". Since I_{BL}/I_{REF} become much larger due to parallel read-out, the read access time is observed to be less than 1ns. As aforementioned, the bit-counting results can be mapped to different number of activated LRS-cells in the corresponding column, hence the I_{BL} with 256 activated LSR-cells represents a sum of 0 in this case (for a 512×512 array size). For the illustration purpose, here we only perform 21 sets of simulation covering bit-counting results from "-20" to "+20". For each set, 10,000 MC points are simulated by Cadence Spectre using TSMC 65nm PDK. Fig. 3(a) shows the sensing pass rate of different bit-counting values, where sensing failures may occur due to the offset. Even with "-20" or "+20" difference in the bit-counting value, the sensing pass rate is less than 80%. As there are 512+512+10=1,034 binary columns in total for the MLP, every 1,034 MC points are randomly selected as one group each time to generate 10,000 groups of offset patterns. Then we perform the inference on MNIST dataset with the generated offset patterns. Fig. 3(b) shows the distribution of the classification accuracy from 10,000 MC runs. The average accuracy is only 15.04%, which is definitely unacceptable. Therefore, we propose to split a large weight matrix into small ones to maintain a good sensing pass rate of CSA.

D. Array Partition for Implementing Arbitrary Network Size

In this section, we analyze different design options with array partition. Firstly, the size of the sub-array is a key design

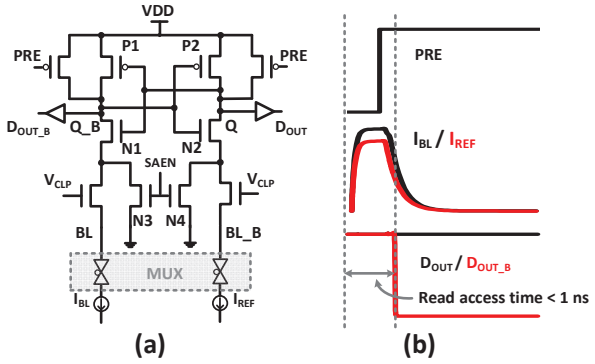

Fig. 2. (a) Schematic of CL-CSA. (b) Simulation waveforms of sensing 40 LRS-cells (BL) against 32 LRS-cells (BL_B). As I_{BL} > I_{REF}, D_{OUT} remains as "1" while D_{OUT_B} drops to "0". Read access time is less than 1ns.

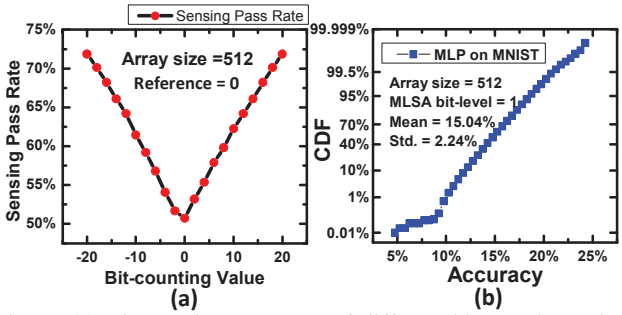

Fig. 3. (a) The sensing pass rate of different bit-counting values when the array size is 512×512. The reference is set as the current for the bit-counting value of 0. (b) The accuracy distribution of MLP on MNIST from 10,000 runs where one single CSA is used as the binary neuron. The average accuracy is only 15.04%.

parameter that may affect the classification accuracy and the cost of system. After the matrix splitting, each small matrix needs to generate a partial sum, which will be added up to obtain the final sum for binary activation. Thus, the precision of the partial sum may affect the value of the final sum and then influence the classification accuracy. As a result, ADC-like MLSAs are employed to generate partial sums with fixed-point precision (larger than 1-bit). To minimize the quantization error of the partial sums, we propose to perform nonlinear quantization where quantization edges (or references) are determined via Lloyd-Max algorithm [16] according to the distribution of the partial sums. For instance, the distribution of the partial sums in the MLP is shown in Fig. 4. 7 quantization edges (or references), and 8 quantization levels acquired from Lloyd-Max algorithm are also annotated. Due to reduced quantization error, nonlinear quantization achieves better accuracy than linear quantization, given the same number of quantization levels. For example, the CNN for CIFAR-10 achieves test accuracy of 86.68% with nonlinear quantization and only 13.90% with linear quantization for 8 quantization levels (or 3-bit MLSA).

A generic system diagram that implements one BNN layer of arbitrary size is shown in Fig. 5 (sub-arrays are assumed to be 64×64). MLSAs in sub-arrays generate digital outputs with fixed-point precision, which then go through a thermometer

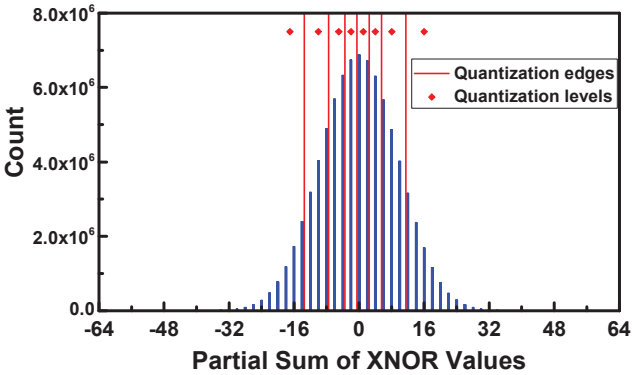

Fig. 4. Distribution of partial sums of XNOR values of the MLP. Sub-arrays are assumed to be 64×64. Red lines are 7 nonlinear quantization edges (or references) and red diamonds indicate 8 quantization levels.

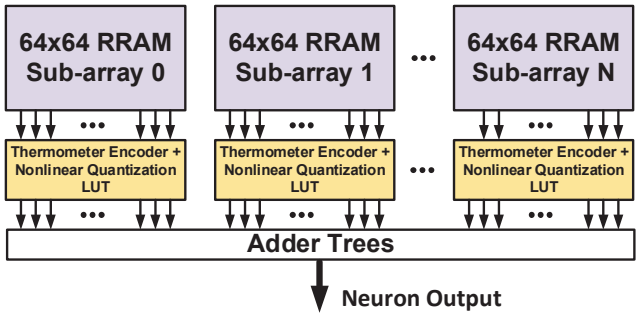

Fig. 5. Generic system diagram for implementing one layer with arbitrary size in a network. The size of sub-array is assumed to be 64×64 as an example.

encoder and look-up table (LUT) to be converted to the corresponding quantization values as partial sums. Adder trees sum up the partial sums to be the final weighted sum, which then goes through the binary activation to generate the neuron output. Here we investigate the cases for the array size of 32×32, 64×64, and 128×128 with MLSA bit-level ranging from 1 to 4 (i.e., 2, 4, 8, 16 quantization levels). The software simulation results for each case are shown in Fig. 6. It can be observed that for both MLP and CNN with 3 different sub-array sizes, the classification accuracy saturates when MLSA bit-level reaches 3-bit. In the meantime, a MLSA bit-level of 2-bit can also provide >98% accuracy (degradation of <1%) for MLP on MNIST when sub-array size is 32×32 or 64×64. Therefore, we select the following three options as benchmarks to analyze the design tradeoffs: (1) option{64, 2}: sub-array size = 64×64, MLSA bit-level = 2; (2) option{64, 3}: sub-array size = 64×64, MLSA bit-level = 3; (3) option{128, 3}: sub-array size = 128×128, MLSA bit-level = 3.

III. Benchmark Results on MNIST and CIFAR-10

A. Considering MLSA Offset and RRAM Variation

In this section, we benchmark the performance of the selected 3 design options on MNIST and CIFAR-10, considering the impact of MLSA offset and RRAM cell resistance variation. The resistance variation is assumed to exhibit Gaussian distribution with a mean of 200KΩ and a standard deviation of 3KΩ. The RRAM LRS resistance distribution could be tightened by the write-verify technique in one-time programming before the inference. The approach

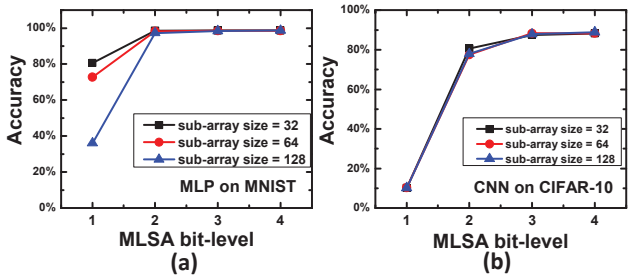

Fig. 6. The classification accuracy of different sub-array sizes and MLSA bit-levels for (a) MLP on MNIST and (b) CNN on CIFAR-10. A 3-bit MLSA is sufficient to provide satisfying accuracy for both evaluations.

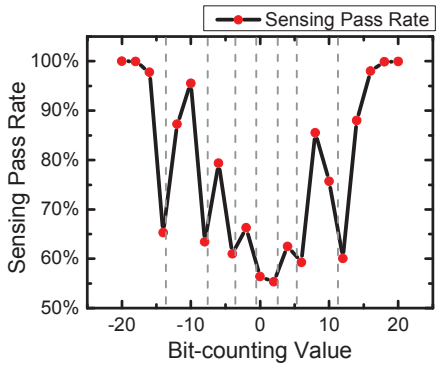

Fig. 7. The sensing pass rate of the bit-counting value in the range of [-20, 20] for the case {64, 3}. Gray dash lines indicate the position of the corresponding 7 sensing references.

mentioned in Section II.C is employed to produce the offset patterns. For each option, 30,000 MC points are generated. Fig. 7 presents the sensing pass rate of the bit-counting value in the range of [-20, 20] for the option {64, 3} as an example. In this case, gray dash lines indicate the position of the corresponding 7 sensing references. The sensing pass rate is relatively low if the bit-counting value is close to a reference due to a small sensing margin. When the difference between the bit-counting value and the reference is large enough, e.g., when bit-counting value is larger than 20 or less than -20 in this case, the pass rate can achieve 100%.

B. Benchmark Results of MLP on MNIST

For the evaluation purpose, here we assume no array-reuse during the inference as all weights are stored on-chip. Thus, different numbers of sub-arrays are needed when the sub-array size varies. Table II summarizes the number of sub-arrays in different layers (except layer #1). Based on 30,000 MC points, 10,000 sets of offset patterns are produced. Then we redo the software statistical simulations with offset information for the selected 3 options. Fig. 8 shows the accuracy distributions from 10,000 MC runs. The option {64, 2} can only achieve an average accuracy of 95.81% while the BNN algorithm accuracy is 98.77%, showing a degradation of 2.96%. Among the 3 options, the option {64, 3} shown in Fig. 8(b) achieves the best average accuracy of 98.56% with a standard deviation of 0.11%, showing only 0.21% degradation compared to the accuracy of ideal BNN algorithm. For the option {128, 3} with the largest sub-array size, the impact of offsets worsened as expected, resulting in a degradation of 0.34%.

TABLE II. NUMBER OF SUB-ARRAYS FOR MLP ON MNIST

Layer #	Matrix Size	Sub-array # 64×64	Sub-array # 128x128
1	784×512	N/A	N/A
2	512×512	64	16
3	512×512	64	16
4	512×10	8	4
Total	N/A	136	36

C. Benchmark Results of CNN on CIFAR-10

As the statistical simulations of CNN on CIFAR-10 are much more time-consuming, we only perform the evaluation for the option {64, 3} and {128, 3}, considering the results of MLP on MNIST in the previous section and software simulation results shown in Fig. 6(b). Table III summarizes the number of sub-arrays for the implementation of the CNN with 6 convolution layers and 3 fully-connected layers. For instance, kernel size of (128, 3, 3, 3) means that the number of output feature map is 128, the number of input feature map is 3, and the filter size is 3×3. The accuracy distribution for two cases from 1,000 MC runs is shown in Fig. 9. The average accuracy of the option {64, 3} and {128, 3} is 86.12% and 86.08%, showing 2.35% and 2.39% degradation respectively compared to the accuracies of ideal BNN algorithm.

IV. Comparison between Sequential and Parallel XNOR-RRAM

We customized a circuit-level macro model NeuroSim [17] that can be used to estimate the area, latency, and energy consumption of hardware accelerators implemented by RRAM synaptic arrays. The hierarchy of the simulator consists of different levels of abstraction from the memory cell parameters and transistor technology parameters, to the gate-level sub-circuit modules, and then to the array architecture.

In this work, we estimated the area, latency, and energy of

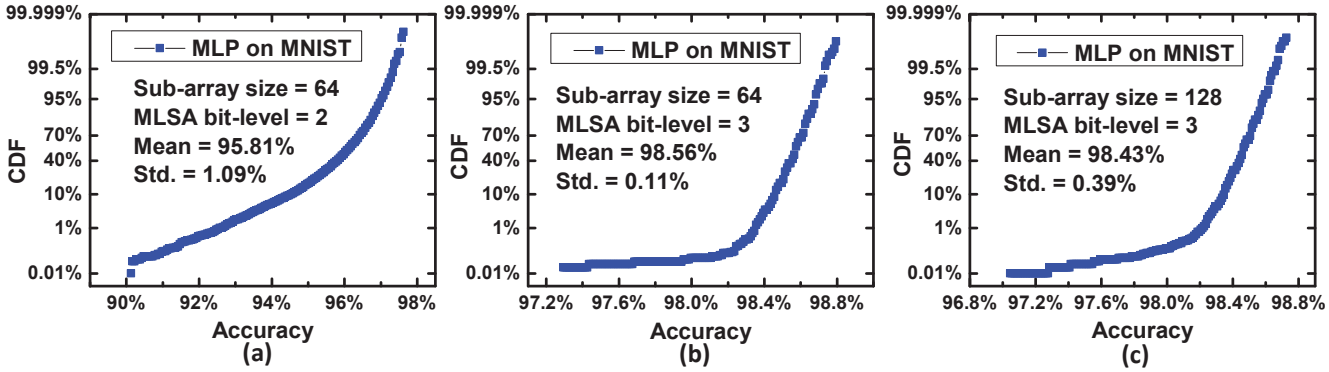

Fig. 8 Classification accuracy distribution of (a) 64×64 sub-array and 2-bit MLSA, (b) 64×64 sub-array and 3-bit MLSA, and (c) 128×128 sub-array and 3-bit MLSA from 10,000 MC runs. The case {64, 3} achieves the best accuracy of 98.56% among 3 cases, showing only 0.21% degradation compared to the accuracy of ideal XNOR-Net.

TABLE III. NUMBER OF SUB-ARRAYS FOR CNN ON CIFAR-10

Layer #	Type	Kernel Size	Sub-array # 64×64	Sub-array # 128x128
1	Conv.	(128, 3, 3, 3)	N/A	N/A
2	Conv.	(128, 128, 3, 3)	36	9
3	Conv.	(256, 256, 3, 3)	72	18
4	Conv.	(256, 256, 3, 3)	144	36
5	Conv.	(512, 256, 3, 3)	288	72
6	Conv.	(512, 512, 3, 3)	576	144
7	FC	(8192, 1024)	2048	512
8	FC	(1024, 1024)	256	64
9	FC	(1024, 10)	16	8
Total	N/A	N/A	3436	863

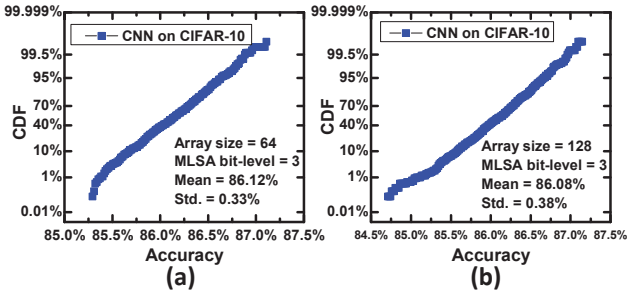

Fig. 9. Classification accuracy distribution of CNN on CIFAR-10 from 1,000 MC runs for the case (a) 64×64 sub-array and 3-bit MLSA and (b) 128×128 sub-array and 3-bit MLSA.

sequential RRAM architecture and parallel XNOR-RRAM architecture (3 options) as discussed in Section III.B for implementing a 256×256 weight matrix. Table IV summarizes the results for each case. As shown in the table, parallel XNOR-RRAM greatly reduces the latency by >350X and the energy-efficiency could be improved by >20X for all 3 options. However, the area overhead is increased, mainly due to the duplicated peripheral circuits used to enable all the sub-arrays in parallel. Considering the tradeoffs among all the metrics, we suggest that the parallel XNOR-RRAM architecture with 128×128 sub-array and 3-bit MLSA could be an optimal design option, which achieves the average accuracy of 98.43% on MNIST (degraded by 0.34%) and 86.08% on CIFAR-10 (degraded by 2.39%), an energy-efficiency of 141.18 TOPS/W (improved by ~33X), and a moderate area overhead (increased by ~1.4X).

V. Conclusion

In this paper, a parallel XNOR-RRAM architecture with a custom bit-cell design is proposed for efficient BNN inference. Array partition is adopted to make it suitable for implementing large-scale BNNs. We analyze the impact of SA offsets on the classification accuracy for 3 design options selected based on the offset patterns generated from Monte Carlo simulations. The estimation of the area, latency, and energy for sequential RRAM with row-by-row read-out and parallel XNOR-RRAM with parallel read-out is performed at 65nm node. Our results show that the design option with 128×128 sub-array size and 3-bit MLSA can achieve an accuracy of 98.43% for MLP on MNIST and 86.08% for CNN on CIFAR-10, showing 0.34% and 2.39% degradation respectively compared to the accuracies of ideal BNN algorithms. If CSA offset cancellation techniques (e.g., switch capacitor sampling in [12]) are employed, we expect a larger sub-array size could be used with less degradation on accuracy. For the current design option for parallel XNOR-RRAM, the estimated energy-efficiency can achieve 141.18 TOPS/W, showing ~33X improvement compared to the conventional sequential RRAM architecture.

TABLE IV. COMPARISON BETWEEN DIFFERENT ARCHITECTURES

Architecture	Area (μm^2)	Latency (ns)	TOPS/W
Sequential RRAM	33,987.7	5036.28	4.23
XNOR-RRAM {64,2}	75,467.7	12.40	157.64
XNOR-RRAM {64,3}	83,196.4	12.70	81.79
XNOR-RRAM {128,3}	46,824.1	13.69	141.18

Acknowledgment

This work is in part supported by NSF-CCF-1552687, NSF-CCF-1652866, NSF-CCF-1715443, NSF-CCF-1740225 and Qualcomm.

References

[1] K. He, et al., "Deep residual learning for image recognition," in *IEEE CVPR*, 2016.

[2] M. Courbariaux, et al., "Binarized neural network: Training deep neural networks with weights and activations constrained to+ 1 or-1," *arXiv: 1602.02830*, 2016.

[3] M. Rastegari, et al., "XNOR-net: ImageNet classification using binary convolutional neural networks," *arXiv: 1603.05279*, 2016.

[4] Y.-H. Chen, et al., "Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *IEEE ISSCC*, 2016.

[5] S. Park, et al., "Neuromorphic speech systems using advanced ReRAM based synapse," in *IEEE IEDM*, 2013.

[6] G. W. Burr, et al., "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element," in *IEEE IEDM*, 2014.

[7] P.-Y. Chen, et al., "Mitigating effects of non-ideal synaptic device characteristics for on-chip learning," in *ACM/IEEE ICCAD*, 2015.

[8] R. Fackenthal, et al., "A 16Gb ReRAM with 200MB/s write and 1GB/s read in 27nm technology," in *IEEE ISSCC*, 2014.

[9] S. Yu, et al., "Binary neural network with 16 Mb RRAM macro chip for classification and online training," in *IEEE IEDM*, 2016.

[10] M. Hu, et al., "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE Transactions on Neural Networks and Learning Systems*, 25 (10), 1864-1878, 2014.

[11] J. Zhang, et al., "A machine-learning classifier implemented in a standard 6T SRAM array," in *IEEE Symp. VLSI Circuits*, 2016.

[12] C.-P. Lo, et al., "Embedded 2Mb ReRAM macro with 2.6 ns read access time using dynamic-trip-point-mismatch sampling current-mode sense amplifier for IoE applications," in *IEEE Symp. VLSI Circuits*, 2017.

[13] L. Ni, et al., "An energy-efficient and high-throughput bitwise CNN on sneak-path-free digital ReRAM crossbar," in *IEEE/ACM ISLPED*, 2017.

[14] T. Tang, et al., "Binary convolutional neural network on RRAM," in *ACM/IEEE ASP-DAC*, 2017.

[15] L. Gao, et al., "Demonstration of convolution kernel operation on resistive cross-point array," *IEEE Electron Device Lett.*, vol. 37, no. 7, 870-873, 2016.

[16] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, 6(1), 7-12, 1960.

[17] P.-Y. Chen, et al., "NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," in *IEEE IEDM*, 2017.