Department: Expert Opinion

Monolithically Integrated RRAM- and CMOS-Based In-Memory Computing Optimizations for Efficient Deep Learning

Shihui Yin
Arizona State University

Yulhwa Kim
Pohang University of Science and Technology

Xu Han
Arizona State University

Hugh Barnaby
Arizona State University

Shimeng Yu
Georgia Institute of Technology

Yandong Luo
Georgia Institute of Technology

Wangxin He
Arizona State University

Xiaoyu Sun
Georgia Institute of Technology

Jae-Joon Kim
Pohang University of Science and Technology

Jae-sun Seo
Arizona State University

***Abstract*—Resistive RAM (RRAM) has been presented as a promising memory technology toward deep neural network (DNN) hardware design, with nonvolatility, high density, high ON/OFF ratio, and compatibility with logic process. However, prior RRAM works for DNNs have shown limitations on parallelism for in-memory computing, array efficiency with large peripheral circuits, multilevel analog operation, and demonstration of monolithic integration. In this article, we propose circuit-/device-level optimizations to improve the energy and density of RRAM-based in-memory computing architectures. We report experimental results based on prototype chip design of 128 × 64 RRAM arrays and CMOS peripheral circuits, where RRAM devices are monolithically integrated in a commercial**

Digital Object Identifier 10.1109/MM.2019.2943047
Date of current version 8 November 2019.

90-nm CMOS technology. We demonstrate the CMOS peripheral circuit optimization using input-splitting scheme and investigate the implication of higher low resistance state on energy efficiency and robustness. Employing the proposed techniques, we demonstrate RRAM-based in-memory computing with up to 116.0 TOPS/W energy efficiency and 84.2% CIFAR-10 accuracy. Furthermore, we investigate four-level programming with single RRAM device, and report the system-level performance and DNN accuracy results using circuit-level benchmark simulator NeuroSim.

■ **Deep learning algorithms** have shown tremendous success in recent years[1] for various applications including computer vision, speech recognition, language translation, etc.

However, an increasing gap exists between the exponential network size growth of state-of-the-art DNNs (e.g., tens of millions of parameters) and the incremental energy-efficiency improvement of conventional memory technologies (e.g., CMOS scaling) for hardware accelerator designs.[2]

To bridge this gap and largely improve the memory energy efficiency, in-memory computing (IMC) has been proposed in recent years across different memory technologies.[3–9] IMC typically asserts multiple or all rows simultaneously to perform multiply-and-accumulate (MAC) computations of DNNs inside the memory, e.g., along the bitlines with analog current/voltage.

SRAM-based IMC works[3–5] demonstrate high-energy efficiency, however typically such IMC SRAM bitcells include a few additional transistors, which degrades density and leakage. In addition, custom peripheral circuits such as analog-to-digital converters (ADCs) incur lower array efficiency. Since one SRAM cell occupies 150–300 F^2 (F is the feature size of a technology node), on-chip SRAMs cannot hold all weights of DNNs. Therefore, CMOS hardware accelerators inevitably involve off-chip DRAMs at the system level, which results in high energy consumption.

Consequently, a number of works have proposed to bring computation closer to the DRAM. DRAM-based near-memory computing proposes to add logic in the DRAM die, however logic capability in the optimized DRAM process is relatively limited. On the other hand, DRAM-based IMC is more challenging, because the conventional 1T1C DRAM read is destructive, and thus requires additional overheads such as data copy and write back.[6] DRAM cell designs with nondestructive read have been proposed (e.g., 2T1C, 3T1C),[7] but they directly degrade density, which is especially disadvantageous for area-efficient DRAMs.

Both SRAM and DRAM are volatile and have increasing concerns on leakage power in scaled CMOS nodes. To that end, resistive nonvolatile memory (NVM) has emerged as a good alternative due to high density, nonvolatility, and nondestructive read. Among several well-known candidates including phase change memory (PCM), resistive RAM (RRAM), and mag- netic RAM (MRAM), this article focuses on RRAM owing to its high ON/OFF ratio, multilevel programmability, and monolithic integration capability.

In addition, both SRAM and DRAM are volatile and have increasing concerns on leakage power in scaled CMOS nodes. To that end, resistive nonvolatile memory (NVM) has emerged as a good alternative due to high density, nonvolatility, and nondestructive read. Among several well-known candidates including phase change memory (PCM), resistive RAM (RRAM), and magnetic RAM (MRAM), this article focuses on RRAM owing to its high ON/OFF ratio, multilevel programmability, and monolithic integration capability.

There has been only a few works that have demonstrated monolithically integrated RRAM and CMOS for DNN hardware design.[8–10]

Mochida *et al.*[8] designed 180-nm and 40-nm prototype chips with embedded RRAM arrays. However, only simple multilayer percepton (MLP) has been demonstrated that resulted in low-inference accuracy of 90.8% for MNIST data set. An RRAM macrointegrated with multilevel sense amplifiers (SAs) in 55-nm CMOS logic process was recently reported by Xue

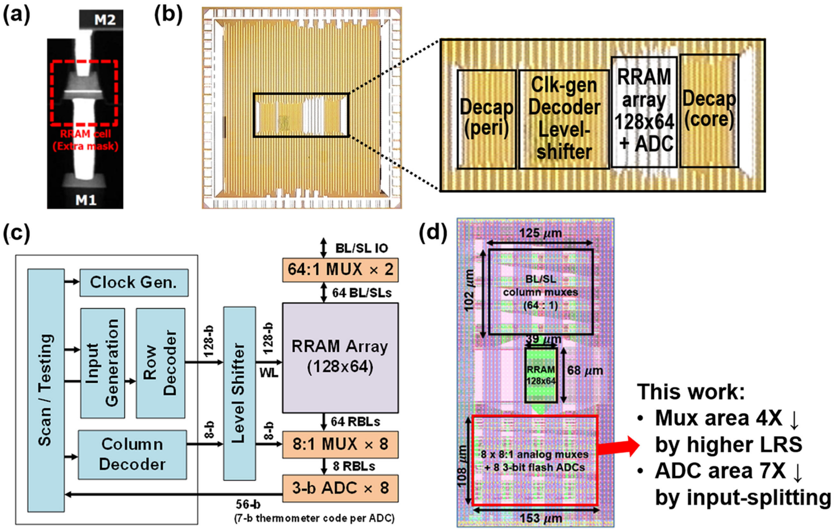


Figure 1. Prototype chip design with monolithically integrated RRAM and 90-nm CMOS technology[14] (adapted with permission). This work presents further energy/area optimization.

et al.,[9] targeting convolutional neural networks (CNNs). However, a relatively low CNN accuracy of 81.83% accuracy for CIFAR-10 data set was achieved with binary/ternary precision. Moreover, only nine WLs are asserted simultaneously in the 256 × 512 subarray, which limits further parallelism, and a relatively complex 4-bit ADC was employed at the RRAM array periphery, degrading array efficiency and energy consumption. In the article by Shulaker *et al.*,[10] a monolithically integrated 3-D nanosystem has been presented, which connects CMOS transistors, carbon nanotube transistors (CNFET), and RRAM devices in different layers with inter-layer vias. A small-scale support vector machine accelerator has been demonstrated, but applicability for larger DNNs has not been shown. While there has been considerable improvement in the CNFET integration with CMOS or RRAM, in terms of manufacturability and yield, integration of RRAM with CMOS in commercial technology is much superior.[11]

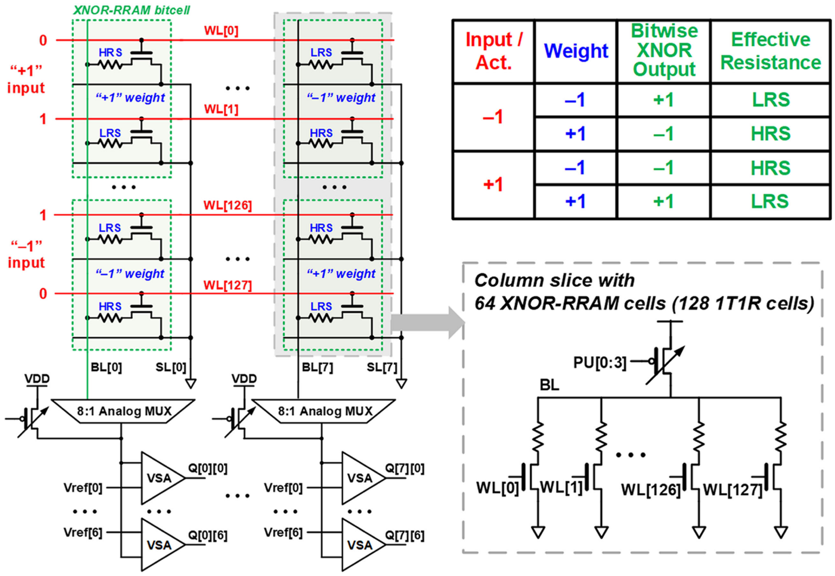
In this article, we address such limitations in RRAM-based IMC toward energy-/area-efficient and accurate DNN hardware design, using monolithic integration of RRAM and CMOS. In particular, we investigate three different device/circuit techniques: 1) modulating resistance values for binary RRAM devices; 2) peripheral circuit minimization with input-splitting technique; and 3) multilevel RRAM programming. We report measurement results of 90-nm CMOS prototype chip in monolithically integrated RRAM arrays, which executes IMC operations of CNNs for CIFAR-10 data set.

In our IMC architecture, monolithic integration of RRAM and CMOS is crucial, since we need dense connections to all wordlines (WLs) and bitlines of the RRAM array. If RRAM and CMOS are not monolithically integrated (e.g., using through-silicon vias or silicon interposers), the bitline and WL delays will be excessive and the integration density will be too low. Furthermore, monolithic integration of RRAM with CMOS is simpler and less expensive than that with CNT.[10] RRAM process is CMOS fabrication compatible, with just a few layers of oxide deposition at the contact via at back-end-of-line (BEOL) compatible temperature. Typically, only one additional mask/lithography is required, allowing RRAM integration to be low cost.

RRAM PROTOTYPE CHIP DESIGN

We designed a prototype chip for RRAM-based robust IMC with Winbond's embedded RRAM technology,[11] which monolithically integrates 90-nm CMOS and RRAM between M1 and M2 [see Figure 1(a)]. Figure 1(b) shows the pad-limited chip micrograph and the core area of the chip. As shown in the top-level block diagram in Figure 1(c), the chip design includes a 128 × 64 1T1R array, row decoder, level shifter, eight 8-to-1 column multiplexers, eight 3-bit flash ADCs based on seven voltage-mode SAs, and two 64-to-1 column decoders for RRAM cell-level programming. The row decoder has two modes of operation: 1) turning on all WL signals simultaneously for binary or low-precision MAC operation; or 2) generating one-hot WL signals for cell-level programming. As shown in Figure 1(d), ADCs and column multiplexers consume a large portion of the core area. In this article, further energy/area optimization is investigated including peripheral circuit minimization by using higher LRS and input-splitting scheme.

Conventional binary RRAMs cannot effectively represent the positive and negative weight values (+1 and −1) in binarized neural networks (BNNs),[12] because the high-resistance state (HRS) and low-resistance state (LRS) values of binary RRAM devices are both positive. In addition, as shown in Figure 2, the activation/weight value combinations of +1/+1 and −1/−1 should result in the same effective resistance. To that end, we proposed to use a "XNOR-RRAM" bitcell design[13,14] for BNNs. As shown in Figure 2, the XNOR-RRAM bitcell is designed with differential RRAM cells and differential WLs. The binary activations are implemented with the differential WLs, and the binary weights are implemented with the HRS/LRS values of XNOR-RRAM bitcells. With all differential WLs asserted simultaneously, all cells in the same column in parallel compute the binary MAC operations. Since one XNOR-RRAM cell consists of two 1T1R bitcells, 128 × 64 1T1R array effectively represents 64 × 64 XNOR-RRAM cells.



Input / Act.	Weight	Bitwise XNOR Output	Effective Resistance
−1	−1	+1	LRS
	+1	−1	HRS
+1	−1	−1	HRS
	+1	+1	LRS

Figure 2. IMC operation of XNOR-RRAM[14] (adapted with permission).

Both the preliminary simulation results[13] and initial measurement results[14] of the XNOR-RRAM design only considered the default LRS and HRS values for the binary RRAM devices, and employed a 3-bit ADC at the periphery for digitizing the analog partial MAC value. In this article, we investigate three further optimizations in monolithically integrated RRAM devices and peripheral circuits, toward enhancing the energy efficiency and density of the RRAM-based IMC systems.

First, since the default LRS value (~6 kΩ) consumes large current and the ON/OFF ratio is relatively high (~150), we explore using higher LRS values (e.g., ~12 and ~24 kΩ) to evaluate the tradeoff between current reduction, ON/OFF ratio, and CNN accuracy.

Second, although a 3-bit ADC is relatively simple, it still consumes a large area compared to the RRAM array itself, resulting in low-array efficiency. We present further algorithm/hardware improvements beyond the previous input-splitting techniques,[15] and employ binary SAs with an unified reference voltage across all columns, instead of ADCs at the RRAM array periphery, for digitizing the analog partial MAC values. Considering that tightly spaced reference voltages make flash ADCs more susceptible to variability at low voltages, we show that the proposed input-splitting scheme actually results in much improved accuracy at lower supplies.

Finally, beyond binary RRAM devices, we investigate four-level programming with the same RRAM devices in our prototype chip, and experimentally validate the density, energy, and performance gains by benchmarking a CNN for CIFAR-10 data set.

HIGHER RESISTANCE FOR LRS DEVICES

In binary RRAM devices, only two states per device exist, namely LRS (high conductance) and HRS (low conductance). In commercial RRAM technologies that are typically used for storage applications, ON/OFF ratio of higher than 100 has been reported. Having a large ON/OFF ratio is certainly good, but on the other hand, having high conductance value for the LRS leads to high current consumption.

To that end, for a given HRS value fixed, and if we have higher LRS values in binary RRAM devices, then the current and energy consumption could be largely reduced. On the other hand, compared to the default LRS, targeting LRS to

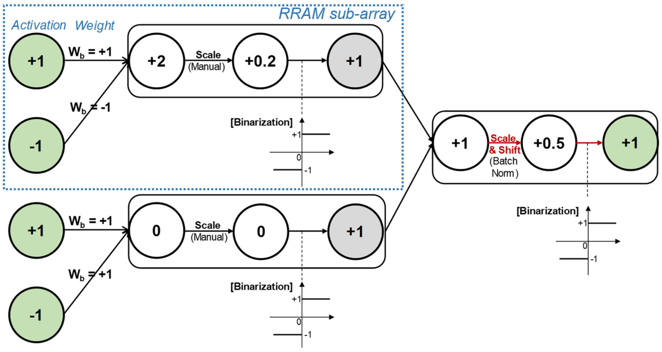


Figure 3. New input-splitting scheme that allows unified reference voltage for all SAs in the RRAM array periphery.

have a higher resistance value can result in wider distribution after programming or more susceptible to nonideal effects such as read disturb. In addition, and if the LRS and HRS ranges become relatively close, it will adversely affect the DNN accuracy for the RRAM-based IMC hardware.

PERIPHERAL CIRCUIT MINIMIZATION WITH INPUT-SPLITTING SCHEME

Input splitting is a method of the BNN architecture design for ADC-free IMC.[15] Input splitting reconstructs a large BNN layer with a network of small layers. It splits input of a large layer so that the number of inputs per split group is less than or equal to row count of the given RRAM array. Each split group constructs a new small layer, and the binary output generated from small layers is accumulated and subsequently binarized with a threshold value of zero. Then, each layer of input-split BNN can fit on RRAM array so that the array can generate binary neuron values as output values. However, batch normalization governs that each neuron has its own threshold value, which necessitates each column to have a digital-to-analog converter,[4] adding a large overhead.

In this article, we modified the conventional input-splitting method[15] to eliminate columnwise threshold values. Batch normalization conducts scaling and shifting operation, and the shifting operation generates threshold values. Therefore, as illustrated in Figure 3, we removed batch normalization before output binarization of small layers. Instead, we experimentally found a proper scaling factor for prebinarization values of small layers. For the RRAM array with 64 rows, we found that, by scaling prebinarization value with 1/20, most of scaled values lie in the range of $[-1, 1]$. As there is no shifting operation on prebinarization value of small layers, the columnwise threshold is fixed to 0. Then, we added batch normalization after the merge to compensate for the loss of batch normalization on small layers.

We tested a VGG-like CNN for CIFAR-10 dataset, which has the network structure of input-128C3-128C3-MP2-256C3-256C3-MP2-512C3-512C3-MP2-1024FC-1024FC-10FC.[12] Here, 128C3-128C3 refers to the convolution layer with 128 input feature maps, 3 × 3 kernels, and 128 output feature maps, MP2 refers to 2 × 2 max-pooling, and 1024FC refers to the fully connected layer with 1024 hidden neurons.

As we used RRAM arrays with 64 effective rows, the input counts per input-split BNN layer was set to 63 for convolution layers and 64 for fully connected layers. We used 63 for convolution layer because we use 3 × 3 kernel for convolution, and 63 is the closest value less than equal to 64. In addition, to make the input of convolution layer be divided by 63, we changed the number of channels to be an integer multiple of 7. Using Torch, we trained the input-split BNN with the same training condition used in conventional input splitting.[15] For comparison, we trained baseline BNN (nonsplit BNN), input-split BNN with columnwise threshold, and input-split BNN without columnwise threshold. The algorithm simulation results showed that the input-split BNN without columnwise threshold model has compatible accuracy (86.64%) with the baseline BNN (88.46%) and input-split BNN with columnwise threshold (88.24%).

MULTILEVEL RRAM DEVICES

Multilevel Programming Scheme

To achieve 2-bit RRAM, two more conductance states are inserted between minimum and maximum conductance levels so that the conductance interval is equal between adjacent states. A write–verify programming scheme is iterated until less than 2% of RRAM cells are outside the target conductance range for each of the four levels. The maximum number of write–verify iterations to program one RRAM cell is specified as $N_{\max}$. For each conductance state, 4096 RRAM cells in the prototype chip are programmed and measured. It is observed that the conductance distribution becomes more concentrated as $N_{\max}$ increases. The $N_{\max}$ to achieve the target conductance range are 15, 30, 15, and 10 for the four conductance states, respectively. After programming, the percentage of the RRAM cells that are outside the target conductance ranges were 0.32%, 1.32%, 0.92%, and 0.44%, respectively.

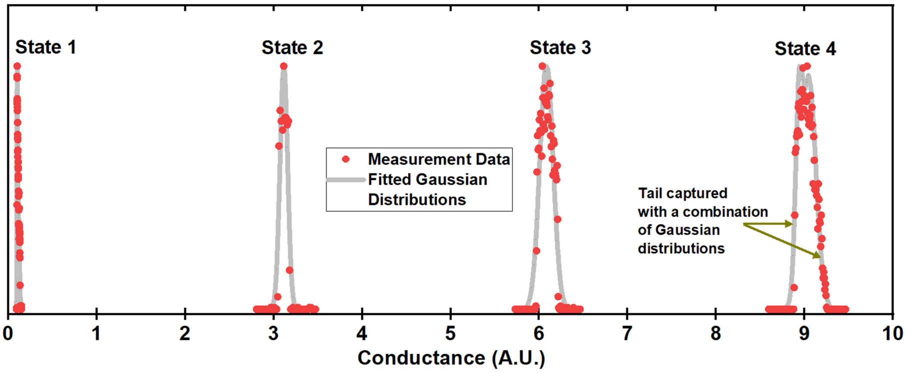


Figure 4. Conductance distribution is shown for four levels of RRAM device programming. Both measurement data from prototype chip and fitted Gaussian distribution curves are shown.

Inference Accuracy Simulation

The inference accuracy for a CNN is simulated with the measured 2-bit RRAM data. However, considering the limited measurement data (4096 data points for each state) compared to the total number of parameters in a CNN, we first fitted the probability density function (PDF) of the measured conductance data with a linear combination of multiple Gaussians as the fitted PDF. Then, the conductance values were generated with the fitted PDF for a large CNN. Figure 4 shows the PDF of the measured conductance and the conductance values generated with fitted PDF. The distribution tails of the experiment data are captured with the fitted PDF.

Using 2-bit weights and 4-bit activations, we benchmarked the same VGG-like CNN for CIFAR-10. It is assumed that each 2-bit weight is stored into one RRAM cell. We first trained the CNN with the quantized training method proposed by Wu *et al.*,[16] and obtained the software baseline accuracy of 91.7%. The 2-bit weights are then mapped to conductance states, where the conductance values of each RRAM cell are generated with the fitted PDFs of the corresponding states. The inference accuracy is simulated for three different array size 64 × 64, 128 × 128, and 256 × 256, where we employed flash ADCs with 5-bit precision using nonlinear quantization.[13]

MEASUREMENT AND SIMULATION RESULTS

Binary-RRAM-Based IMC Energy and Accuracy Characterization with Higher LRS and Input Splitting

We envision that large binary CNNs are mapped onto multiple RRAM arrays, where weights for different input channels are stored on different rows, weights for different output channels are stored on different columns, and weights within each convolution kernel (e.g., $9 = 3 \times 3$) are stored in different RRAM macros.[3,14] Subsequently, the partial MAC results from different RRAM macros are accumulated via digital simulation. In the article by Yin *et al.*,[14] 3-bit ADC was used to digitize the analog partial MAC values, where seven reference voltages for each flash ADC required offset cancellation in order to achieve $>83\%$ CIFAR-10 accuracy. The new input-splitting scheme presented in this article substantially reduces such calibration overhead, since we only need binary SAs to digitize the analog partial sum, and the same reference voltages are used for all 64 columns of the RRAM array.

Another important challenge for the flash ADC is that the adjacent reference voltages are very close to each other, especially since the partial sum data distribution is concentrated near zero.[13] If we lower the supply voltage, the reference voltages actually become even closer to each other, which makes it more susceptible to process variation. On the other hand, since the input-splitting scheme allows us to have

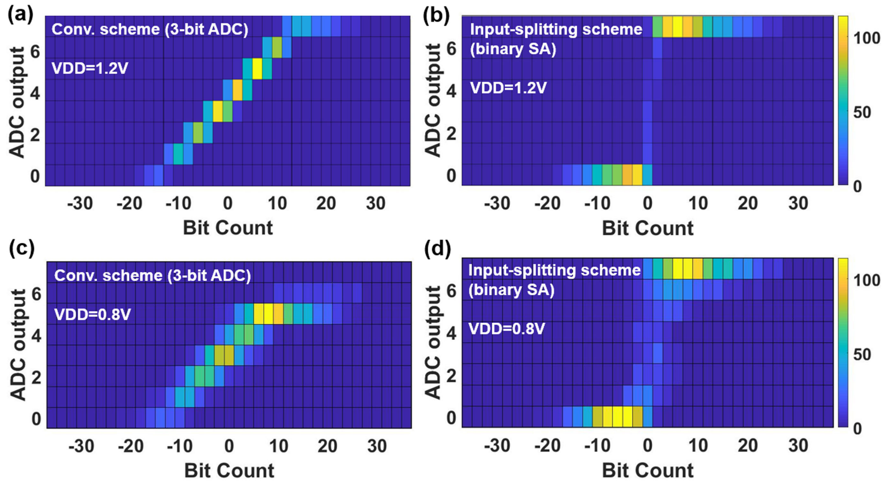


Figure 5. Measured ADC output results compared with bitcount values from BNN algorithm.

only one reference voltage for the SAs, the digitization is inherently more robust to variability and noise.

We performed chip measurements for the experiments of higher LRS values and the input-splitting scheme. For the higher LRS experiment, we programmed RRAM devices with different target LRS values of 6, 12, and 24 kΩ. For the input-splitting scheme experiment, with the same XNOR-RRAM prototype chip, we only use one SA out of the seven SAs that are present in the flash ADC. This means that when we employ the input-splitting scheme, the overall ADC area in the RRAM macro is effectively reduced by 7×.

In Figure 5, we show the comparison of the bitcount values from the BNN algorithm (i.e., ideal partial sum values) and the measured ADC output values using 12-kΩ LRS target, for both the conventional scheme with 3-bit ADCs and the input-splitting scheme with binary SAs. As we compare the 1.2- and 0.8-V supply results, it can be seen that the ADC output values become less accurate at 0.8 V. However, with a single reference level, the input-splitting scheme still maintains more robust operation even at lower voltages.

As shown in Figure 6, the energy efficiency (TOPS/W) of RRAM-based IMC increases with higher LRS values and with lower supply voltages. The CIFAR-10 accuracy values for the VGG-like CNN with voltage scaling are reported in Figure 7 with different LRS values for both the input-splitting scheme with binary SAs and the conventional scheme with 3-bit ADCs. For the conventional scheme with ADCs, it can be seen that the CIFAR-10 accuracy degrades by a large amount when the supply voltage scales below the nominal 1.2 V. This is due to the fact that the seven reference voltages for the flash ADC are separated only by a small voltage value, which aggravates with lower supply voltages, incurs more ADC output errors, and adversely affects the DNN accuracy.

For the input-splitting scheme, there is only one reference voltage used by all eight SAs for 64 columns; the SA operation is more robust against voltage scaling, noise, and variability.

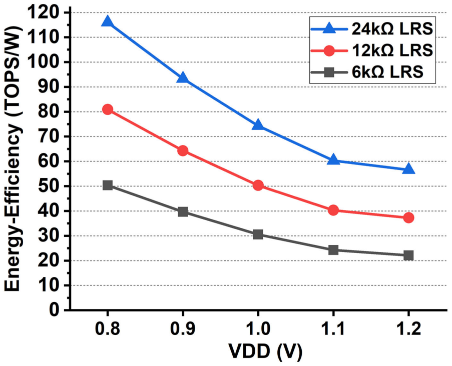


Figure 6. Energy efficiency with voltage scaling for RRAM IMC with different LRS values.

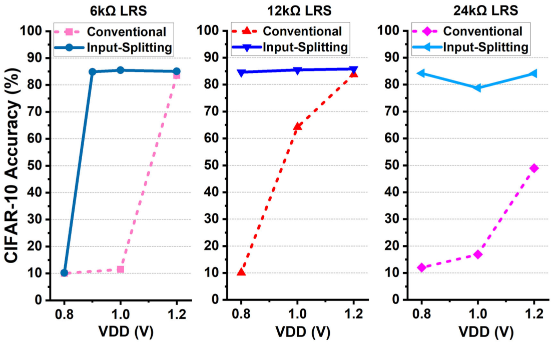


Figure 7. CIFAR-10 accuracy with voltage scaling for RRAM IMC with different LRS values and input-splitting scheme.

As a result, Figure 7 shows that high CNN accuracy is maintained for the input-splitting scheme for 12-kΩ/24-kΩ LRS values, down to 0.8-V supply. The input-splitting scheme also shows higher accuracy for cases when RBL voltage is around 0.6–0.7 V (high gain region for SA with differential NMOS input) for bitcount values near zero, e.g., higher supply with 6-kΩ LRS and lower supply with 24-kΩ LRS value.

The conventional scheme[14] achieves energy efficiency of 20.8 TOPS/W at 1.2-V supply (see Figure 6), while achieving 83.5% accuracy with binary CNN. Jointly optimizing the use of higher LRS value of 12 kΩ (24 kΩ), the proposed input-splitting scheme effectively enabled voltage scaling down to 0.8 V without any additional accuracy loss, improving the energy-efficiency by 3.7× (5.2×), and achieving 80.9 (116.0) TOPS/W.

Table 1. CNN simulation results with 2-bit RRAM for different RRAM array sizes.

RRAM array size	64 × 64	128 × 128	256 × 256
CIFAR-10 accuracy	91.2%[1]	89.2%[1]	79.0%[1]
	(91.4%)[2]	(89.3%)[2]	(81.2%)[2]
Chip area (mm^2)	19.64	19.21	14.71
Read dynamic energy (layer-by-layer, μJ)	33.73	32.50	55.46
Leakage energy (μJ)	1.78	0.77	0.81
Latency (ms)	8.90	6.11	12.93
Energy efficiency (TOPS/W)	17.35	18.52	10.95
Throughput (FPS)	112.38	163.72	77.36

[1] Accuracy with ADC quantization and RRAM conductance variation.
[2] Accuracy with ADC quantization (without RRAM conductance variation).

Two-Bit RRAM-Based CNN Accelerator Performance Benchmarking with Neurosim

Two-bit RRAM could further increase the integration density for the CNN accelerator. The performance benchmarking for 2-bit RRAM-based CNN accelerator is conducted in NeuroSim,[17] where the aforementioned VGG-like CNN is utilized. We assume that eight columns share one ADC in the RRAM array, and there are a total of eight ADCs in the RRAM array periphery. The inference computation is processed layer by layer. Table 1 presents the benchmarking results with different RRAM array sizes.

First, the inference accuracy drops as the array size is increased, since the ADC precision is fixed at 5 bits. This is attributed to the fact that the partial sum distribution becomes broader with larger array size, and, therefore, quantization loss is increased. It should be noted that the conductance variation of 2-bit RRAM only leads to small accuracy drop when comparing the accuracy with ADC quantization only and with both ADC quantization and RRAM conductance variation.

In terms of chip area, 256 × 256 array shows smaller chip area compared with 128 × 128 array due to the increased array efficiency. However, only small chip area increase is observed when array size is reduced to 64 × 64. Comparing with 128 × 128 array, for 256 × 256 array, chip area is reduced as less subarrays are needed. It can be explained by the fact that in 64 × 64 array, the periphery circuit size is reduced due to lower maximum column partial sum current, therefore, the array efficiency does not drop significantly compared with 128 × 128 array.

For the read latency and dynamic energy consumption, comparing with 128 × 128 array, 64 × 64 array needs more partial sum accumulations between subarrays, which leads to higher latency and energy consumption. For 256 × 256 subarray, the large column current leads to significantly higher ADC energy consumption and, therefore, the overall energy consumption is increased. Besides, the larger column partial sum current leads to larger transmission gate (TG) size in the multiplexer, which induces higher latency for the decoder to drive the TG gate capacitor.

CONCLUSION

In this article, we demonstrated RRAM-based IMC with 90-nm CMOS prototype chips that monolithically integrated RRAM and CMOS in different vertical layers. Using device-/circuit-/algorithm-level techniques, both the energy efficiency and density of binary RRAM-based IMC hardware improved substantially, achieving up to 116.0 TOPS/W and 84.2% accuracy for CIFAR-10 data set. Experiments with 2-bit RRAM demonstrate sufficient separation between four conductance levels, and show higher CNN accuracy up to 128 × 128 RRAM array size.

ACKNOWLEDGMENT

The authors would like to thank Winbond Electronics for chip fabrication support. This work was supported in part by NSF-SRC-E2CDA under Contract 2018-NC-2762B; by the National Science Foundation (NSF) under Grant 1652866, Grant 1715443, and Grant 1740225; in part by JUMP C-BRIC and ASCENT programs (SRC programs sponsored by DARPA); and in part by the Nano-Material Technology Development Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT (NRF-2016M3A7B4910249).

REFERENCES

1 Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

2 X. Xu *et al.*, "Scaling for edge inference of deep neural networks," *Nature Electron.*, vol. 1, pp. 216–222, 2018.

3 Z. Jiang *et al.*, "XNOR-SRAM: In-memory computing SRAM macro for binary/ternary deep neural networks," in *Proc. IEEE Symp. VLSI Technol.*, 2018, pp. 173–174.

4 H. Valavi *et al.*, "A 64-tile 2.4-Mb in-memory-computing CNN accelerator employing charge-domain compute," *IEEE J. Solid-State Circuits*, vol. 54, pp. 1789–1799, Jun. 2019.

5 X. Si *et al.*, "A twin-8 T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning," in *Proc. IEEE Int. Solid-State Circuit Conf.*, 2019, pp. 396–398.

6 V. Seshadri *et al.*, "Ambit: In-memory accelerator for bulk bitwise operations using commodity DRAM technology," in *Proc. IEEE/ACM Int. Symp. Microarchit.*, 2017, pp. 273–287.

7 S. Li *et al.*, "DRISA: A DRAM-based reconfigurable in-situ accelerator," in *Proc. IEEE/ACM Int. Symp. Microarchit.*, 2017, pp. 288–301.

8 R. Mochida *et al.*, "A 4 M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture," in *Proc. IEEE Symp. VLSI Technol.*, 2018, pp. 175–176.

9 C.-X. Xue *et al.*, "A 1 Mb multibit ReRAM computing-in-memory macro with 14.6 ns parallel MAC computing time for CNN Based AI edge processors," in *Proc. IEEE Int. Solid-State Circuit Conf.*, 2019, pp. 388–390.

10 M.-M. Shulaker *et al.*, "Three-dimensional integration of nanotechnologies for computing and data storage on a single chip," *Nature*, vol. 547, pp. 74–78, 2017.

11 C. Ho *et al.*, "Integrated HfO2-RRAM to achieve highly reliable, greener, faster, cost-effective, and scaled devices," in *Proc. IEEE Int. Electron Devices Meeting*, 2017, pp. 2.6.1–2.6.4.

12 I. Hubara *et al.*, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

13 X. Sun *et al.*, "XNOR-RRAM: A scalable and parallel resistive synaptic architecture for binary neural networks," in *Proc. Design, Autom. Test Eur. Conf. Exhib.*, 2018, pp. 1423–1428.

14 S. Yin *et al.*, "High-throughput in-memory computing for binary deep neural networks with monolithically integrated RRAM and 90 nm CMOS," 2019, *arXiv:1909.07514*. [Online]. Available: https://arxiv.org/abs/1909.07514.

15 Y. Kim *et al.*, "Input-splitting of large neural networks for power-efficient accelerator with resistive crossbar memory array," in *Proc. IEEE Int. Symp. Low Power Electron. Design*, 2018, Article no. 41.

16 S. Wu *et al.*, "Training and inference with integers in deep neural networks," 2018, *arXiv:1802.04680*. [Online]. Available: https://arxiv.org/abs/1802.04680.

17 P. Chen *et al.*, "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *IEEE Trans. Comput.-Aided Design Integr. Circuit Syst.*, vol. 37, no. 12, pp. 3067–3080, Dec. 2018.

Shihui Yin is currently working toward the PhD degree at the School of Electrical, Computer and Energy Engineering, Arizona State University. He is a student member of IEEE. Contact him at: Shihui.Yin@asu.edu.

Yandong Luo is currently working toward the PhD degree at the School of Electrical and Computer Engineering, Georgia Institute of Technology. He is a student member of IEEE. Contact him at: yandongluo@gatech.edu.

Yulhwa Kim is currently working toward the PhD degree at the Department of Creative IT Engineering, Pohang University of Science and Technology. She is a student member of IEEE. Contact her at: yulhwa.kim@postech.ac.kr.

Wangxin He is currently working toward the PhD degree at the School of Electrical, Computer and Energy Engineering, Arizona State University. He is a student member of IEEE. Contact him at: wangxinh@asu.edu.

Xu Han is currently working toward the PhD degree at the School of Electrical, Computer and Energy Engineering, Arizona State University. She is a student member of IEEE. Contact her at: xhan37@asu.edu.

Xiaoyu Sun is currently working toward the PhD degree at the School of Electrical and Computer Engineering, Georgia Institute of Technology. He is a student member of IEEE. Contact him at: xiaoyusun@gatech.edu.

Hugh Barnaby is a professor in the School of Electrical, Computer and Energy Engineering, Arizona State University. His research interests include device physics and modeling, microelectronic device/sensor design and manufacturing, and analog/RF/mixed-signal circuit design. He is a Fellow of IEEE. Contact him at: hbarnaby@asu.edu.

Jae-Joon Kim is a professor in the Department of Creative IT Engineering, Pohang University of Science and Technology. His research interests include neuromorphic circuit and system, low power VLSI design, and flexible device/circuit design. He is a member of IEEE. Contact him at: jaejoon@postech.ac.kr.

Shimeng Yu is an associate professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology. His research interests are nanoelectronic devices and circuits for energy-efficient computing systems. He was a recipient of the NSF CAREER Award in 2016, the IEEE Electron Devices Society (EDS) Early Career Award in 2017, and the Semiconductor Research Corporation (SRC) Young Faculty Award in 2019. He is a senior member of IEEE. Contact him at: shimeng.yu@ece.gatech.edu.

Jae-sun Seo is an assistant professor in the School of Electrical, Computer and Energy Engineering, Arizona State University. His research interests include energy-efficient hardware design for machine learning and neuromorphic computing. He received the IBM Outstanding Technical Achieved Award in 2012 and the NSF CAREER Award in 2017. He is a senior member of IEEE. Contact him at: jaesun.seo@asu.edu.