Received 9 April 2021; revised 9 June 2021; accepted 27 June 2021. Date of publication 29 June 2021; date of current version 12 July 2021.
The review of this article was arranged by Editor S. Menzel.

Digital Object Identifier 10.1109/JEDS.2021.3093478

Efficient and Optimized Methods for Alleviating the Impacts of IR-Drop and Fault in RRAM Based Neural Computing Systems

CHENGLONG HUANG [1], NUO XU [2], KENI QIU [3], YUJIE ZHU [3], DESHENG MA [1], AND LIANG FANG [1]

1 Institute for Quantum Information & State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha 410073, China
2 College of Computer, National University of Defense Technology, Changsha 410073, China
3 College of Information Engineering, Capital Normal University, Beijing 100089, China

CORRESPONDING AUTHOR: N. XU (e-mail: oun_ux@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1003304; in part by the National Natural Science Foundation of China under Grant 61832007; and in part by the Research Foundation from NUDT under Grant ZK20-02.

ABSTRACT Resistive switching random access memory (RRAM) shows its potential to be a promising candidate as the basic in-memory computing unit for deep neural networks (DNN) accelerator design due to its non-volatile, low power, and small footprint properties. The RRAM based crossbar array (RRAM CBA) is usually employed to accelerate DNN because of its intrinsic characteristic of executing multiplication-and-accumulation (MAC) operation according to Kirchhoffs' law. However, some major non-ideal effects including IR-drop and Stuck-at-Faults (SAF) in real RRAM CBA are typically ignored in the DNN accelerator design because of the considerations of training speed and design closure. Such non-ideal effects will conduct the variations of output column current and voltage and further cause serious degradation in computing accuracy. Thus, direct mapping from the weights of DNN model without considering the IR-drop and SAF to RRAM CBA is unrealistic. In this work, two efficient and optimized methods including adding additional tunable RRAM row and Trans-impedance amplifier (TIA) based RRAM are proposed to recover the computation accuracy with a view to reducing variation of the output column current of the RRAM CBA and output voltage of the TIA in each column respectively. The two optimized methods are evaluated in different sizes of RRAM CBA and different resistance levels of RRAM cell. The simulation results show that the two optimized methods can further suppress the degradation of computing accuracy induced by IR-drop and SAF for LeNet-5 with the MNIST dataset and VGG16 with the CIFAR-10 dataset.

INDEX TERMS Neural computing systems, RRAM based crossbar array, IR-drop, stuck-at-faults, hardware optimization methods.

I. INTRODUCTION

In the big-data era, there is an ever-increasing requirement for the performance improvement of data processing. Nevertheless, for data-intensive applications, the memory wall bottleneck hinders the development of traditional von Neumann architecture with higher performance [1].

Deep neural networks (DNN) techniques provide effective solutions to some data-intensive applications such as image classification [2], [3], object detection [4], [5], and semantic segmentation [6], [7]. However, the key multiplication-and-accumulation (MAC) operations in the DNN are time-consuming and require lots of energy to run in the conventional von Neumann processors. New technologies, e.g., RRAM are promising to act as the hardware platform for speeding up the DNN based computing process and other logic operations [8], [9]. This is because of their capabilities to offer processing-in-memory (PIM). The key component of the RRAM-based DNN accelerator

is an RRAM crossbar array (CBA) in which the MAC operations are physically performed using analog physical characteristics [10]. Recently, the RRAM-based DNN accelerators are also widely studied as an efficient technique to address the performance bottlenecks of conventional processors such as energy consumption and latency [11], [12], [13]. There are however several non-ideal effects in the array-level, e.g., IR-drop [14], [15] or cell-level, e.g., Stuck-at-Fault (SAF) [16], [17]. Such effects introduce limitations on the computational accuracy of the RRAM-based DNN accelerator. The IR-drop induced by the wire resistance results in non-uniform voltage and current distribution in the RRAM CBA [18]. The SAF (including Stuck-At-0 (SA0) and Stuck-At-1 (SA1)) caused by the cell failure (corresponding to the device get stuck at low resistance state (LRS) and high resistance state (HRS)) also damages the expected weight pattern [19]. Both of the above effects reduce the computational accuracy if the ideal DNN is directly mapped to the realistic RRAM CBA.

Several recent research works proposed techniques to address the above issues. These works are based on either software or hardware compensation techniques for recovering the accuracy degradation caused by the device and circuit non-ideal effects. For instance, a software compensation technique is proposed by B. Liu *et al.* where the IR-drop effect is compensated by alleviating the adverse effect of the wire resistance and the sneak-path in a large RRAM CBA [15]. In their design, the training algorithm is accordingly modified to obtain the new RRAM resistance state that generates a weight matrix closest to the ideal target weights due to the existence of the IR-drop. The new RRAM resistance state compensates for the difference from the ideal target state caused by the IR-drop. Liu *et al.* also proposed retraining and remapping algorithms to compensate for the computation error caused by the defects of the RRAM device [20]. Furthermore, Chen *et al.* proposed an accelerator-friendly training method that leverages the inherent self-healing capability of DNN to map the large weights to the normal RRAM cells. The experimental results show that their method gains significant improvements in a large-scale RRAM CBA with the clustered SAF [17]. However, these works either require retraining with the modified loss function or need computationally expensive operations for the deployment of a neural network accelerator based on RRAM CBA.

An example of hardware compensation techniques is the work by He *et al.*, where they propose a framework called PytorX. This framework evaluates two ideas of digital SAF error correction and noise injection adaption training method for IR-drop and fault compensation [21]. However, they did not provide the hardware design for the error correction. Soudry *et al.* also programmed and then tuned in further iterations on the chip according to the differences between the ideal output and actual output [22]. However, this method also needs to program the device multiple times, which shortens the device lifetime. Jain and Raghunathan

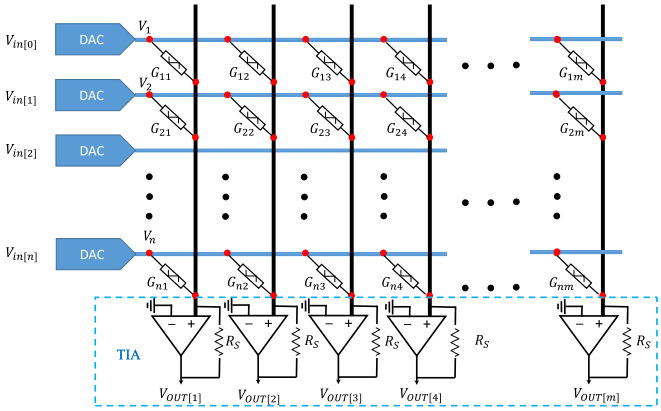


FIGURE 1. N×M RRAM CBA for MAC operation.

further proposed the compensation hardware which consists of registers and multiplier to mitigate dynamic and hardware instance-specific errors [23]. In this paper, we propose two efficient and optimized hardware methods to suppress the shift of the output current and voltage. This addresses the accuracy loss problems caused by the IR-drop and SAF without retraining the neural network. Our hardware compensation techniques consist of RRAM cells instead of the registers and multiplier in [23]. The proposed methods also offer low-overhead due to the superior performance of the RRAM. Our main contributions in this work are as follows:

- We propose a simple optimization method of adding a tunable RRAM row in the RRAM CBA to compensate for the output current deviation in the CBA. It can reduce the accuracy loss induced by the reduction of the current, for example, the IR-drop only, SA1 only, and the coexistence of IR-drop and SA0 in the RRAM CBA.
- A novel Trans-impedance amplifier is proposed based on the RRAM (RRAM-TIA). Using this RRAM-TIA suppresses the output voltage deviation in the TIA hence alleviates the accuracy loss caused by the IR-drop and SAF.
- The simulation results on LeNet-5 and VGG16 on the MNIST and CIFAR-10 datasets validate the effectiveness of the proposed techniques.

The rest of this paper is organized as the following. In Section II, we briefly introduce the RRAM CBA, IR-drop, and SAF. Then in Section III, we present the proposed hardware optimization schemes. Simulation results are presented in Section IV followed by conclusions in Section V.

II. PRELIMINARIES

Fig. 1 shows a typical structure of the passive RRAM CBA which is often used to perform the MAC operation in an RRAM-based DNN accelerator. The RRAM CBA utilizes the analog characteristics of a single RRAM device to perform MAC operations. There are also some peripheral circuits such as analog-to-digital (ADC), digital-to-analog (DAC), and shift-and-add. The input feature maps are fed into the

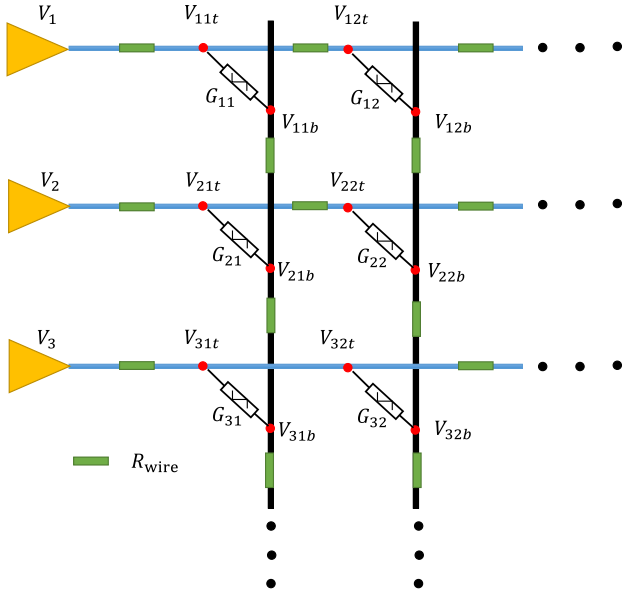


FIGURE 2. The circuit schematic of RRAM CBA with IR-drop.

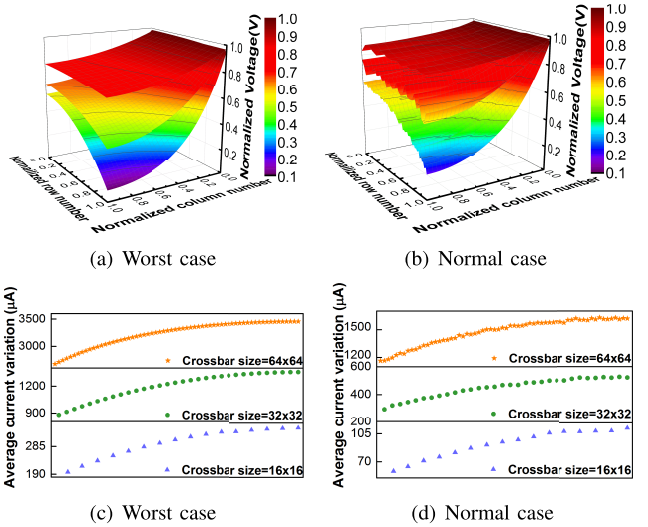


FIGURE 3. Distribution of Voltage drop of CBA and average output current variation at each column of CBA with different dimensions. Surfaces from top to down are 16×16, 32×32, 64×64.

DAC to obtain input voltage across the RRAM cells. The current is summed at the CBA output and further fed into the TIA, ADC, and shift-and-add to generate the output feature maps. The relationship between the input and output voltage in the RRAM CBA is:

$$V_{\text{out}[m]} = -\sum_{i=1}^{n} V_i \times G_{\text{im}} \times R_s \tag{1}$$

where G_{im} is the conductance of RRAM cell and R_s is a sense resistor in TIA. The original computation of the convolution and fully connected layer in the DNN is based on the weighted summation with input feature maps. This can be then implemented in the RRAM CBA as in Fig. 1.

A. IR-DROP AND SAF

Along with the shrinking of the feature size, the metal wires in the chip become thinner. This leads to an increase in the resistance per unit length. The ITRS2015 [24] reports that the wire resistance between the adjacent junctions at 10 nm technology node exceeds 20Ω. Therefore, in practice, the wire resistance should be considered in the RRAM CBA that is abstracted into the network composed of RRAM cells and wire resistance. The voltage distribution in the CBA is also nonuniform because of the IR-drop problem. This is in contrast with the assumption made based on the ideal conditions. Fig. 2 shows the equivalent circuit schematic of the RRAM CBA considering the wire resistance. In this case, the resistance of the wire caused by the IR-drop reduces the voltage between each RRAM cell beyond the expected value that is calculated based on the scheme of Fig. 1. Therefore, the real output current and the output voltage of the TIA at each column are also less than the expected value.

In practice, the output current at the RRAM CBA in Fig. 2 is

$$I_{\text{out}[m]} = \sum_{i=1}^{n} (V_{\text{imt}} - V_{\text{imb}}) \times G_{\text{im}} \tag{2}$$

where V_{imt}, V_{imb} are the voltages of the top terminal and bottom terminal of a RRAM cell respectively. In Fig. 2, R_{wire} is the wire resistance that causes V_{imt} to be lower than that of V_i. We use the components in Fig. 2 to generate the HSPICE netlist. In the simulation, we examine different RRAM CBA sizes 16×16, 32×32, 64×64. The wire resistance is set to 25Ω and the input voltage is 1.2V. The resistance of RRAM is set in the range from 10KΩ to 1MΩ.

The parameters we adopt here for the RRAM CBA are as the same as in [25] which are also considered in Section IV. To investigate the impact of the CBA caused by the IR-drop, in Fig. 3 we plot the distribution of the voltage drop and output average current deviation at each column of the CBA under the worst and normal cases. The simulations are conducted using the HSPICE. The worst-case in Fig. 3 denotes the case where the resistance of RRAM is 10KΩ. For the normal case in Fig. 3, the resistance of RRAM is a random value in the range between 10KΩ to 1MΩ.

The output current deviation at each column are presented in Figs. 3(c) and 3(d) which indicate the difference between ideal condition and real condition with the IR-drop. By increasing the CBA, the current deviation is increased which results in lower accuracy in the RRAM-based DNN accelerator. Nevertheless, in accelerator design the non-ideal effect created by the IR-drop is typically ignored, see, e.g., [13].

The SAF causes the RRAM cell to get stuck at a state so it is not changed by an external stimulus. There are two categories of such states including SA0 and SA1 which are corresponding to the LRS and HRS respectively. Due to the

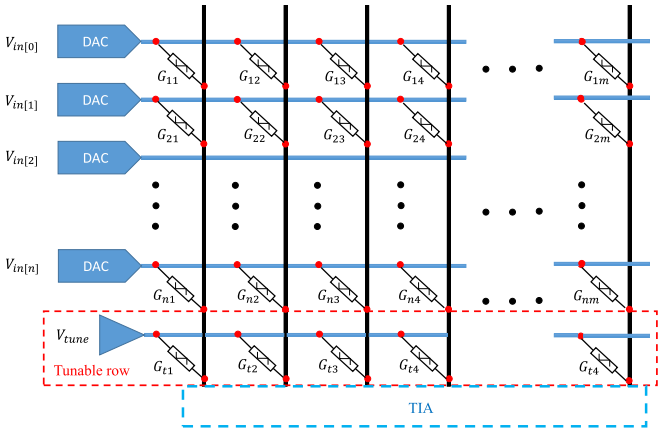


FIGURE 4. The schematic of additional tunable RRAM row.

SAF, the weights are not correctly mapped to the RRAM cell. The mismatch between the expected conductance of the RRAM cells (after mapping) and the real conductance of RRAM cells with SAF. results in the deviation of the output current of the column (and also the output voltage of TIA) in comparison with the expected situation after mapping. It should be noted that the SAF often exists in a real RRAM CBA because of the material defects in the manufacturing process and the limited endurance of the RRAM cell. The SAF can be detected by the memory test as a random distribut [18].

III. PROPOSED METHODOLOGY

As discussed in Section II, the current deviation of the output column in the CBA are directly caused by the non-ideal effects including the IR-drop and SAF. The current deviation also cause deviation of the output voltage of TIA and degrade the computation accuracy. Therefore, alleviating/compensating the current or voltage deviation is required to effectively reduce the loss of accuracy. Here, we propose two methods: adding tunable RRAM, row, and changing the traditional TIA to RRAM-TIA to deal with the deviation of output current or voltage.

A. ADDITIONAL TUNABLE RRAM ROW

Due to the IR-drop, the actual output current of the column is less than the ideal situation. Therefore, introducing an additional current can solve this problem. In this method, we propose to add a tunable RRAM row to the original CBA as a source to provide the additional current (Fig. 4). In general, the current deviation between the ideally theoretical value (without considering the non-ideal effects) and real current is

$$\Delta I_{\text{out}[m]} = I_{\text{ideal}-\text{out}[m]} - I_{\text{out}[m]} = \Delta I(\mu, \sigma) \tag{3}$$

The current difference, $\Delta I(\mu, \sigma)$, is adaptively compensated by the RRAM cell in the additional tunable RRAM row by tuning the input voltage and conductance of the RRAM cell in the tunable row. Here we employ a dynamic CBA solver proposed in [21] to determine the resistance of tunable RRAM row.

Based on equation (1), the actual weight matrix represented by a CBA with conductance state G_{im} is not ideal, the actual conductance state is $G^*_{\text{im}} = f(R_{\text{im}}, R_{\text{wire}})$, where R_{wire} is wire resistance due to IR-drop, and R_{im} represent the resistance state of RRAM cell. The actual output current of the column is, therefore,

$$I_{\text{out}[m]} = -\sum_{i=1}^{n} V_i \times G^*_{\text{im}} \tag{4}$$

The additional tunable RRAM row provides the additional current

$$I_{\text{tune}[m]} = V_{\text{tunable}} \times G_{tm} \tag{5}$$

where V_{tunable} is the input voltage of additional tunable RRAM row, and G_{tm} is the conductance of RRAM cell in additional tunable RRAM row. Note that V_{tunable} and G_{tm} are determined using equations (3),(4),(5) as:

$$G_{\text{im}}, V_{\text{tunable}} = \min_{G_{\text{im}}, V_{\text{tunable}}} || I_{\text{ideal}-\text{out}[m]} - I_{\text{out}[m]} - I_{\text{tune}} ||^2 \tag{6}$$

The detailed process of determining the resistance of tunable RRAM row comprises of the following four steps: (1) We employ the dynamic solver to obtain the ideal average output current of the RRAM CBA. (2) We then add the tunable RRAM row to the original RRAM CBA in (1) and consider the existence of the IR-drop and SAF. The resistance of the tunable RRAM row is initiated by an initial value. (3) We further employ the dynamic solver to obtain the real average output current of RRAM CBA in (2). (4) Finally, the resistance of the tunable RRAM row and the input voltage are tuned until the current value in (3) becomes very close to the current obtained in (1).

Note that this method (adding a tunable row in RRAM CBA) is not only suitable for alleviating the impact of IR-drop but also suitable for the SA1. This is because the SA1 also reduces the current of the column current. Nevertheless, this method is not applicable to reduce the impact of SA0, as the SA0 increases the column currently. In practice, the CBA may include multiple non-ideal effects, and different effects may have opposite impacts on the column current. For instance, the IR-drop and SA0 may decrease the column current while the column current is increased in the single SA0 case. Therefore, the method of adding a tunable row might be also useful for real CBA which includes multiple non-ideal effects because of the offset of different impacts. This is later discussed in the results section.

B. TRANS-IMPEDANCE AMPLIFIER BASED RRAM

The additional tunable row only deals with the decreasing column current. It is therefore necessary to find a way to deal with the cases where the column current is increased. For example, if only the SA0 fault exists in the CBA, the

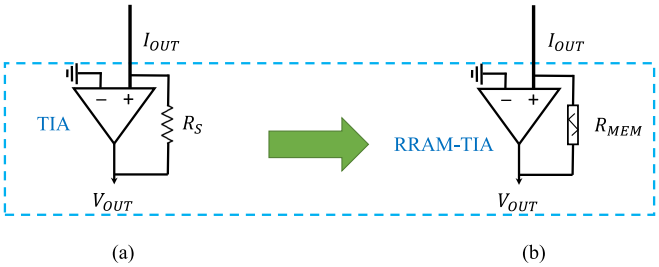


FIGURE 5. The schematic of TIA and RRAM-TIA.

column current will be increased. Therefore, we develop an extra method to deal with the non-ideal effects by blocking the voltage deviation of TIA. Fig. 5 shows this method where a special TIA based on RRAM (RRAM-TIA) is proposed to enable adjustable gain for the TIA. In traditional TIA, as it is shown in Fig. 5(a), the output voltage is the multiplication of output current and the fixed value resistor. A TIA with the tunable gain is designed by replacing the fixed value resistor with a tunable RRAM cell [26], [27], as in Fig. 5(b). Using this RRAM-TIA, the voltage deviation induced by the non-ideal effects is compensated by tuning the resistance of the RRAM. The detailed explanations are below.

The ideal voltage is

$$V_{\text{ideal}-\text{out}[m]} = -I_{\text{ideal}-\text{out}[m]} \times R_s \tag{7}$$

This ideal voltage is our expected value without considering the non-ideal effects of IR-drop and SAF. However, due to the IR-drop and SAF, the real output voltage with IR-drop and SAF is,

$$V_{\text{out}[m]} = -I_{\text{out}[m]} \times R_s \tag{8}$$

From (7) and (8), the deviation of the output voltage is

$$\Delta V_{\text{out}} = |I_{\text{out}[m]} - I_{\text{ideal}-\text{out}[m]}| \times R_s \tag{9}$$

Replacing the TIA with the RRAM-TIA, (9) is rewritten as

$$\Delta V_{\text{out}} = |I_{\text{out}[m]} \times R_{\text{MEM}} - I_{\text{ideal}-\text{out}[m]} \times R_S| \tag{10}$$

$$\Delta V_{\text{out}} = R_s \times |I_{\text{out}} \times \frac{R_{\text{MEM}}}{R_s} - I_{\text{ideal}-\text{out}[m]}| \tag{11}$$

Based on the (11), the ΔV_{out} can be minimized by changing the value of R_{MEM}. This reduces the deviation caused by IR-drop and SAF. The ideal output current I_{ideal} and the real current I_{out} of each column are also obtained using the value of R_s in the TIA. The value of R_{MEM} of the RRAM-TIA is then obtained to minimize ΔV_{out}.

IV. SIMULATION AND RESULT

In this work, we use the recognition results for the LeNet-5 for MNIST(60000 pictures for training, 10000 pictures for validation) and VGG16 for the CIFAR-10(50000 pictures for training, 10000 pictures for validation) dataset to verify the validity of the two proposed methods. All the works are based on the open-source framework, PytorX [21], including the LeNet-5 model and the models of the ideal and non-ideal

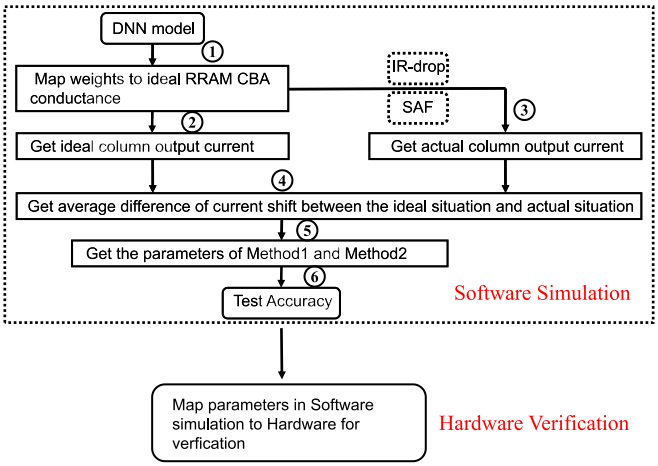


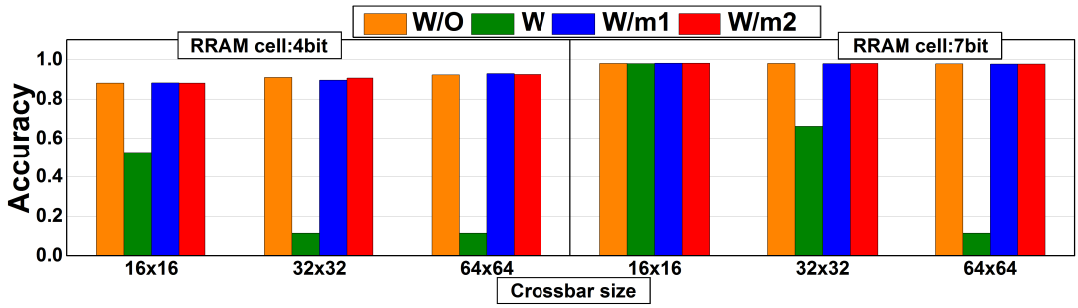
FIGURE 6. The overall flow of the proposed methods.

RRAM CBA. We further experiment on the VGG16 with CIFAR-10 based on the PytorX [21]. The PytorX is a comprehensive framework that performs end-to-end training, mapping, and evaluation for a crossbar-based neural network accelerator. We take the LeNet-5 as an example to explain the process of our simulation.

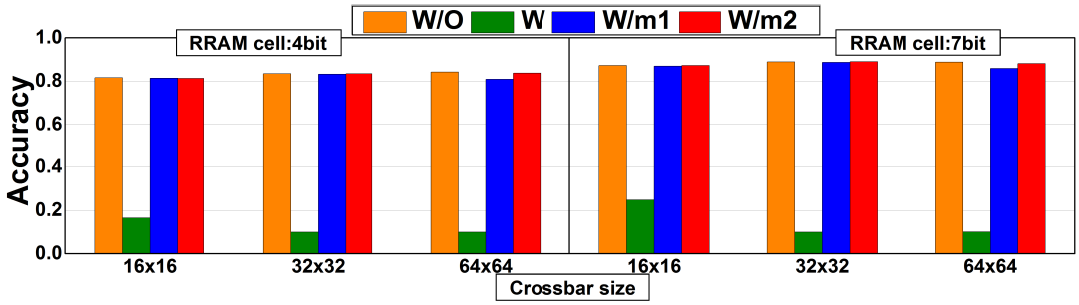
The simulation comprises six steps shown in Fig 6: (1) We train the LeNet-5 without considering non-ideal effects during training. Then, the trained model is mapped on the CBA in Pytorx; (2) We can collect the CBA output current shift with or without non-ideal effects for all of the testing data; (3) The non-ideal effects including IR-drop and SAF are added in Simulation. The actual column output current can be gotten after simulation; (4) We then calculate the average difference of the current shift between the ideal and actual situations; (5)The resistance values of each RRAM in the additional row or RRAM-TIA are then obtained based on the results of step (2) to compensate the current or voltage differentials in each column; (6) We calculate the average accuracy of RRAM CBA after optimization for all of the testing data with considering non-ideal effects. The simulation is performed on different sizes of RRAM CBA including 16×16, 32×32, 64×64. The resolutions of ADC and DAC are 8 bits. The number of the resistance states of the RRAM cell used in the CBA array is set to 7 and 4, respectively. The simulation is performed on an Nvidia RTX-2080Ti GPU.

A. RESULTS ON IR-DROP

Fig. 7 shows the recognition results in the ideal case, the case with the IR-drop, the case with the IR-drop and method 1, and the case with the IR-drop and method 2. These cases are represented by W/O, W, W/m1, W/m2 in Fig. 7. The simulation results show that the impact of IR-drop on the computation accuracy is increased with the size of CBA. The two optimization methods of adding tunable RRAM row and RRAM-TIA also significantly reduce the accuracy degradation in all cases. It is also seen that the IR-drop impact on the computation accuracy induced is highly sensitive to the resistance levels, the size of the network, and the size



(a) The impact of IR-drop and optimization with 4bit and 7bit RRAM cells(LeNet-5 with MNIST)



(b) The impact of IR-drop and optimization with 4bit and 7bit RRAM cells(VGG16 with CIFAR-10)

FIGURE 7. The simulation results with considering IR-drop only on different size of RRAM CBA with 4bit and 7bit RRAM cells (Blue bars: Optimization method 1; Red Bars: Optimization method 2).

of the dataset. The results in Fig. 7 also indicate that our proposed methods provide higher accuracy under different resistance levels, network size, and datasets.

B. RESULTS ON IR-DROP AND SAF

The two proposed methods are further evaluated in the cases with IR-drop and SAF. Fig. 8 shows the accuracy with and without the two proposed methods for the IR-drop and different percentages of SAF. The percentage of SA0 is changed from 0 to 10% and the percentage of SA1 is changed from 0 to 40% for the LeNet-5 with MNIST. We also change the percentage from 0 to 10% for VGG16 with CIFAR-10. It is seen in Fig. 8(a), Fig. 8(c), Fig. 8(e), and Fig. 8(g) that the degradation of accuracy is more significant along with the increase of the size of the CBA (the yellow line in each figure). This suggests that the effect of IR-drop is dominant when it coexists with 10% SA0 (which conducts the current of column increase). Smaller CBAs often have higher accuracy because of the lower impact of the IR drop. The accuracy is also improved by increasing the defect rate of SA0 because of the compensation effect (SA0 increases the output current hence compensates for the reduced output current caused by the IR-drop).

Both of the proposed methods can further reduce the degradation of accuracy conducted by IR-drop as it is (green lines and blue lines) shown in Figs. 8(a), 8(c), 8(e), and 8(g). Furthermore, the results in Fig. 8 also confirm that the two methods proposed are efficient in recovering the accuracy in cases where the IR-drop and SA1 coexist (see Figs. 8(b), 8(d), 8(f), and 8(h). To further examine the ability of the two proposed methods, in Fig. 9 and Fig. 10 we present the results for the case with coexisting non-ideal effects SA0, SA1, and the IR-drop. As it is seen, the two optimization methods are effective in this case.

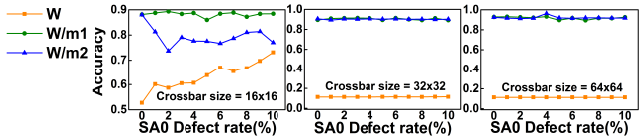
TABLE 1. Experimental results on MNIST and CIFAR-10 dataset (64×64).

Methods	Non-ideal effect			Accuracy degradation(%)	
	SA0(%)	SA1(%)	IR-drop	MNIST	CIFAR-10
[21]	1.75	9	YES	1.01	/
W/m1	2	10	YES	0.1	0.12
W/m2	2	10	YES	0.2	0.4

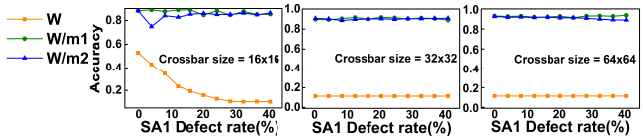
The above results confirm that the two optimization methods are useful in reducing the accuracy loss in the RRAM-based DNN accelerator. Table 1 also compares the results of this work with that of in [21]. It is seen that our proposed methods perform well and are competitive with the state-of-the-art. In addition, the two proposed methods in this work do not require retraining whereas the method proposed in [21] requires retraining of the network which is often a costly process.

V. CONCLUSION

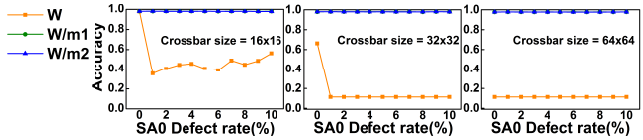
In this work, we propose two efficient methods for improving the computation accuracy created by the non-ideal effect including IR-drop and SAF. The proposed methods do not require retraining and reduce the deviation of column current or voltage of TIA. We propose adding a tunable row to the original RRAM CBA. This compensates for the output current deviation from the ideal condition. Furthermore, we



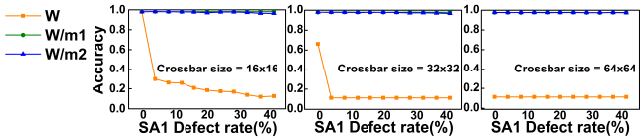
(a) The impact of IR-drop and SA0 with 4bit RRAM cells(LeNet-5 with MNIST)



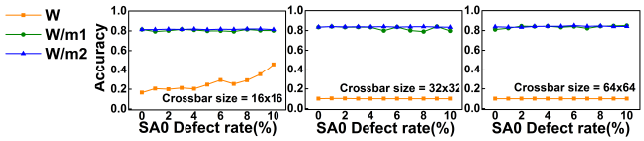
(b) The impact of IR-drop and SA1 with 4bit RRAM cells(LeNet-5 with MNIST)



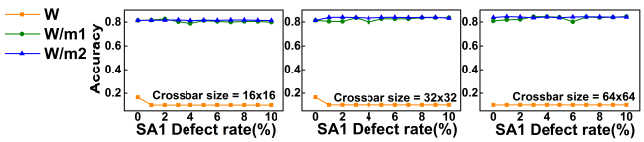
(c) The impact of IR-drop and SA0 with 7bit RRAM cells(LeNet-5 with MNIST)



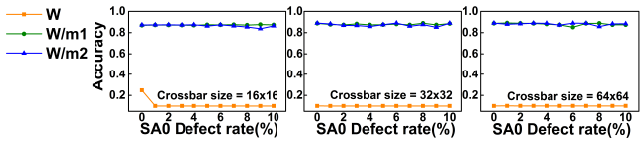
(d) The impact of IR-drop and SA1 with 7bit RRAM cells(LeNet-5 with MNIST)



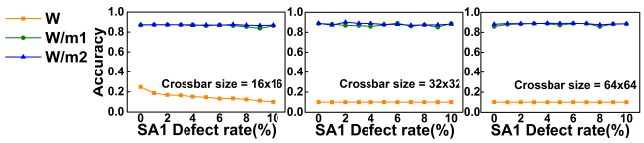
(e) The impact of IR-drop and SA0 with 4bit RRAM cells(VGG16 with CIFAR-10)



(f) The impact of IR-drop and SA1 with 4bit RRAM cells(VGG16 with CIFAR-10)



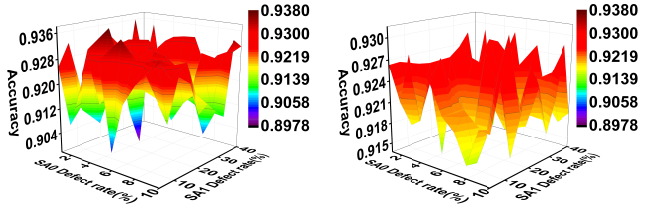
(g) The impact of IR-drop and SA0 with 7bit RRAM cells(VGG16 with CIFAR-10)



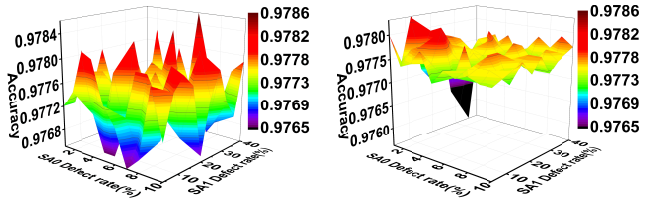
(h) The impact of IR-drop and SA1 with 7bit RRAM cells(VGG16 with CIFAR-10)

FIGURE 8. The simulation results with considering IR-drop and SA0 or SA1 on different size of RRAM CBA with 4bit and 7bit RRAM cells.

propose an RRAM-TIA with tunable gain to compensate for the output voltage deviation. The experimental results demonstrate that the two proposed methods are effective in reducing the accuracy degradation in a real CBA array. In the future, one may consider cases where the potential resistance

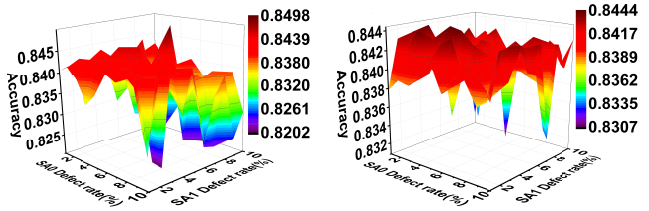


(a) 4bit RRAM cells with method 1 (b) 4bit RRAM cells with method 2

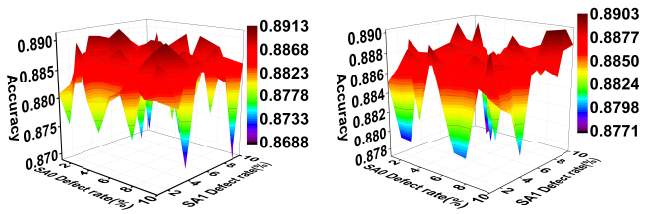


(c) 7bit RRAM cells with method 1 (d) 7bit RRAM cells with method 2

FIGURE 9. The simulation results on LeNet-5 with considering IR-drop and SAF and after optimization on different size of RRAM CBA with 4bit and 7bit RRAM cells(4bit:[W/O:0.9220, W:0.1135], 7bit:[W/O:0.9781, W:0.1135]).



(a) 4bit RRAM cells with method 1 (b) 4bit RRAM cells with method 2



(c) 7bit RRAM cells with method 1 (d) 7bit RRAM cells with method 2

FIGURE 10. The simulation results on VGG16 with considering IR-drop and SAF and after optimization on different size of RRAM CBA with 4bit and 7bit RRAM cells(4bit:[W/O:0.8425, W:0.0998], 7bit:[W/O:0.8883, W:0.1006]).

is changed due to the accumulated current change and other non-ideal effects in the RRAM CBA.

ACKNOWLEDGMENT

The authors thank the calculation support of State Key Laboratory of High Performance Computing, National University of Defense Technology.

REFERENCES

[1] M. M. Waldrop, "The chips are down for Moore's law," *Nature*, vol. 530, no. 7589, p. 144, 2016.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, 2012, pp. 1097–1105.
[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jul. 2017.
[5] B. Singh and L. Davis, "An analysis of scale invariance in object detection-SNIP," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3578–3587.
[6] S. Kim, J.-H. Hong, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," 2019. [Online]. Available: arXiv:1805.11360.
[7] T. Hu, P. Yang, C. Zhang, G. Yu, Y. Mu, and C. G. Snoek, "Attention-based multi-context guiding for few-shot semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8441–8448.
[8] N. Xu, K. J. Yoon, K. Kim, L. Fang, and C. S. Hwang, "Fully functional logic-in-memory operations based on a reconfigurable finite state machine using a single memristor," *Adv. Electron. Mater.*, vol. 4, no. 11, 2018, Art. no. 1800189.
[9] N. Xu *et al.*, "A stateful logic family based on a new logic primitive circuit composed of two antiparallel bipolar memristors," *Adv. Intell. Syst.*, vol. 2, no. 1, 2020, Art. no. 1900082.
[10] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nat. Nanotechnol.*, vol. 15, no. 7, pp. 529–544, 2020.
[11] A. Shafiee *et al.*, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Archit. (ISCA)*, 2016, pp. 14–26.
[12] P. Chi *et al.*, "PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory," in *Proc. ISCA*, 2016, pp. 27–39.
[13] L. Song, X. Qian, H. Li, and Y. Chen, "PipeLayer: A pipelined ReRAM-based accelerator for deep learning," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, 2017, pp. 541–552.
[14] Y. Zhu, X. Zhao, and K. Qiu, "Insights and optimizations on IR-drop induced sneak-path for RRAM crossbar-based convolutions," in *Proc. 25th Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2020, pp. 506–511.
[15] B. Liu *et al.*, "Reduction and IR-drop compensations techniques for reliable neuromorphic computing systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2014, pp. 63–70.
[16] F. Merrikh-Bayat, M. Prezioso, B. Chakrabarti, I. Kataeva, and D. B. Strukov, "Memristor-based perceptron classifier: Increasing complexity and coping with imperfect hardware," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, 2017, pp. 549–554.
[17] L. Chen *et al.*, "Accelerator-friendly neural-network training: Learning variations and defects in RRAM crossbar," in *Proc. Design Autom. Test Europe Conf. Exhibition (DATE)*, 2017, pp. 19–24.
[18] C. Wang *et al.*, "Cross-point resistive memory," *ACM Trans. Design Autom. Electron. Syst.*, vol. 24, no. 4, pp. 1–37, 2019.
[19] L. Xia *et al.*, "Stuck-at fault tolerance in RRAM computing systems," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 1, pp. 102–115, Mar. 2018.
[20] C. Liu, M. Hu, J. P. Strachan, and H. Li, "Rescuing memristor-based neuromorphic design with high defects," in *Proc. 54th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, 2017, pp. 1–6.
[21] Z. He, J. Lin, R. Ewetz, J.-S. Yuan, and D. Fan, "Noise injection adaption: End-to-end reram crossbar non-ideal effect adaption for neural network mapping," in *Proc. 56th ACM/IEEE Design Autom. Conf. (DAC)*, 2019, pp. 1–6.
[22] D. Soudry, D. D. Castro, A. Gal, A. Kolodny, and S. Kvatinsky, "Memristor-based multilayer neural networks with online gradient descent training," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2408–2421, Oct. 2015.
[23] S. Jain and A. Raghunathan, "CxDNN: Hardware-software compensation methods for deep neural networks on resistive crossbar systems," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 6, pp. 1–23, 2019.
[24] *International Technology Roadmap for Semiconductors 2.0: Executive Report*, ITRS Group, London, U.K., 2015.
[25] K. Qiu, W. Chen, Y. Xu, L. Xia, Y. Wang, and Z. Shao, "A peripheral circuit reuse structure integrated with a retimed data flow for low power rram crossbar-based CNN," in *Proc. Design Autom. Test Europe Conf. Exhibition (DATE)*, 2018, pp. 1057–1062.
[26] T. A. Wey and W. D. Jemison, "Variable gain amplifier circuit using titanium dioxide memristors," *IET Circuits Devices Syst.*, vol. 5, no. 1, pp. 59–65, 2011.
[27] T. Wey and W. Jemison, "An automatic gain control circuit with TiO_2 memristor variable gain amplifier," *Analog Integr. Circuits Signal Process.*, vol. 73, no. 3, pp. 663–672, 2012.