Receptive-Field and Switch-Matrices Based ReRAM Accelerator with Low Digital-Analog Conversion for CNNs

Yingxun Fu[1], Xun Liu[1], Jiwu Shu[2], Zhirong Shen[3], Shiye Zhang[1], Jun Wu[1], and Li Ma[1,*]
[1]College of Information Science, North China University of Technology, Beijing, China
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3]Computer Science Department, Xiamen University, Xiamen, China
[*]Corresponding Author: mali@ncut.edu.cn
mooncape@hotmail.com, 1363937238@qq.com, shujw@tsinghua.edu.cn
zhirong.shen2601@gmail.com, zangxy_2020@qq.com, WuJune000@outlook.com

***Abstract*—Process-in-Memory (PIM) based accelerator becomes one of the best solutions for the execution of convolution neural networks (CNN). Resistive random access memory (ReRAM) is a classic type of non-volatile random-access memory, which is very suitable for implementing PIM architectures. However, existing ReRAM-based accelerators mainly consider to improve the calculation efficiency, but ignore the fact that the digital-analog signal conversion process spends a lot of energy and executing time. In this paper, we propose a novel ReRAM-based accelerator named Receptive-Field and Switch-Matrices based CNN Accelerator (RFSM). In RFSM, we first propose a receptive-field based convolution strategy to analyze the data relationships, and then gives a dynamic and configurable crossbar combination method to reduce the digital-analog conversion operations. The evaluation result shows that, compared to existing works, RFSM gains up to 6.7x higher speedup and 7.1x lower energy consumption.**

***Index Terms*—CNN, PIM, Receptive-Field, Crossbar, Switch-Matrices**

I. Introduction

Convolutional neural network (CNN) is one of the most successful branches of deep neural networks (DNN) [2]. As the structures of CNNs become more complex and the amount of vector-matrix operations increases, the impact of memory wall [8] on the processing efficiency of CNN becomes more prominent. Since storage access speed is much lower than computation speed in today's computer architectures, the most important problem is to improve the storage access performance. Some researches propose DNN accelerators based on memristors [1] [4] that could directly process the convolutional computations in memory. Resistive random access memory (ReRAM) is a classic type of non-volatile random-access memory with storage capacity and high throughput for dot-product operation execution [9], which is very suitable for implementing process-in-memory (PIM) architectures.

Since ReRAM-based crossbars utilize analog signals rather than digital signals for computing, a lot of energy and calculation resources are exhausted in the digital-analog signal conversions. Most of existing PIM-based accelerators, such as ISAAC [1], mainly focus on optimizing the pipeline efficiency with fixed-size crossbars as computing units, but they do not well optimize the consumption of signal conversions, which is another important metric for PIM-based accelerators. Focus on these problems, we propose a new ReRAM-based accelerator termed Receptive-Field and Switch-Matrices based CNN Accelerator (RFSM). Our contributions can be summarized as follows.

- To the best of our knowledge, this is the first work restructuring the crossbars to transmit the data among convolutional layers without digital-analog conversions.
- Our study indicates that 1) The input and output data among different convolutional layers could be transmitted in analog signals, directly; 2) based on receptive-field, we can analyze the relationship between the input/output data among different layers, and thus develop the optimization methods, to accelerate the calculation process; 3) The crossbar scale could be resized by switch matrices, which makes the computation scale of ReRAM-based accelerator reconfigurable.
- We propose a novel Receptive-Field and Switch-Matrices based CNN Accelerator termed RFSM, in order to improve the calculation efficiency and to reduce the energy consumption for CNN convolutional operations. The evaluation result shows that, RFSM gains up to 6.7x calculation efficiency and reduces up to 7.1x energy consumption compared to existing ReRAM-based accelerators.

II. Background and Our Motivation

A. Background

Receptive-Fields: In CNN, the sliding filters of each convolutional layer will repeatedly select one area to calculate the convolutional outputs. The selected areas are the receptive-fileds for the outputs at the present convolutional layer. The calculation expression is $I_{a\times b} \times W_{b\times c} = O_{a\times c}$, where I, W and O are matrices. I_i is the receptive-field of O_i.

ReRAM-based Crossbars: Resistive random access memory (ReRAM) is a memristor which can not only store data, but also perform calculation operations. ReRAM cells are usually

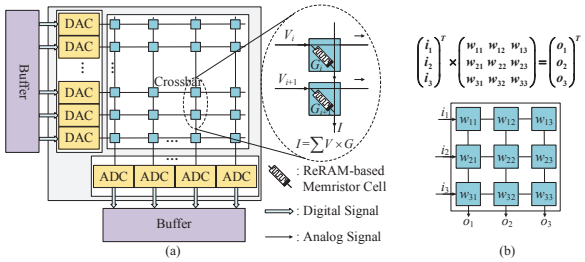


Fig. 1. The receptive-field and ReRAM-based crossbar. (a) The organization and calculation process of ReRAM-based crossbars. (b) The mapping of vector-matrix multiplication in a crossbar.

built as many crossbars for storage and in-situ analog calculation in most ReRAM-based accelerators, such as ISAAC [1], LerGAN [5]. Because of in-situ analog calculations, ReRAM-based crossbars contain a lot of digital-to-analog converters (DAC) and analog-to-digital converters (ADC), as shown in Figure 1 (a). Based on Ohm's law and Kirchhoff law, multiplication operations could be performed at every ReRAM cells, in which a voltage pass through a cell to generate a electric current. All electric currents on the same bitline merge into a new current. Therefore, a ReRAM-based crossbar can calculate a vector matrix multiplication, as shown in Figure 1 (b).

B. Our Motivation

For CNN convolutional layers, existing PIM-based works mainly focus on improving the calculation efficiency, but ignore a fact that the digital-analog conversions occupy a lot of execution time and energy consumption. Fortunately, the receptive-field and switch matrices inspire us that the size of the computation unit could be dynamically reconfigured, and the input and output data in different convolutional layers have some potential relationships. These observations motivate us to design a new architecture to reduce the digital-analog conversions and improve the computation efficiency.

III. The Design of RFSM

A. The Architecture

Focus on the problems referred above, we propose a novel ReRAM-based accelerator termed Receptive-Field and Switch-Matrices based CNN Accelerator (RFSM). Figure 2 gives the architecture. As shown in Figure 2 (a), the key components of RFSM contain one IO interface, one controller, and some computation tiles. Each tile contains an eDRAM buffer, two digital-to-analog registers (DARs), an analog-to-digital register (ADR), some converters (DACs/ADCs), and a crossbar array set, as shown in Figure 2 (b). The eDRAM buffer stores the data flow, DACs/ADCs convert the signals between the analog and digital forms, in order to adapt the vector operations for ReRAM-based computations.

The crossbar array set is the most important component in RFSM, which provides dynamic and configurable computation units for convolutional operations. Figure 2 (c) shows the details of crossbar array set. The crossbar array set is composed of many crossbars, a switch controller, a switch matrices, a series of analog maxpooling circuits and several clippers. The switch controller could transform the switch matrices status to adjust their connections, which gives options to adjust the working cells, and thus satisfies the different computation scales. The clippers could implement activation functions with limiting the bounds of outputs. In RFSM, we can transmit the analog signals from the present layer (output) to the next layer (input) directly, since the effective computation cells are reconfigurablein RFSM. The analog signal direct transmissions effectively reduce the DA/AD conversions, and thus enhance the whole performance.

B. The Key Technologies

Receptive-Field Based Convolution Strategy: The convolutional operations in one layer generate the output data sequentially, but the next layer do not use the data (from the present layer) in keeping with the output order. In RFSM, since the analog signals could be directly transmitted between n adjacent layers, we can define the different priority to change the order of computations based on the receptive-field, in order to shorten the waiting time of the computations of the next layer. The key steps executed by the controller (as shown in Figure 2 (a)) are as follows.

- The controller first calculates the center coordinators, size, and stride of the receptive-fields in the first layer. Existing work [12] gives the calculation equations. Then the controller sends control signals to all switch-matrices controllers for configuring switch-matrices status in crossbar array sets. Please note that the number of layers is configurable. E.g., for each time, we could perform 3-layer convolutions without analog-digital conversions, in order to avoid analog noise impacting the accuracy.
- The controller reads one receptive-field related data set referred above to the input buffer, and then does the convolutional operations in the crossbar array set. Once the outputs have been ready, the controller will send control signals to ADR and OB, in order to transmit the outputs to the eDRAM buffers of the tiles that will be utilized for the next layer.
- The controller reads another receptive-field related data set from eDRAMs to input buffers (according to above parameters such as center coordinate and strides), and repeats above process until all calculation have been done.

Figure 3 gives an example of 2-layer convolutions, in which $b_{1,1}$, $b_{1,2}$, $b_{2,1}$, $b_{2,2}$ could be computed in parallel and directly output the results with analog signal. The switch-matrices referred above will adjust its status to adapt for the referred analog signal inputs, and thus saving the analog-digital conversions.

Dynamic and Configurable Combinations for Crossbars: Each computation tile contains a number of crossbars that have fixed size of ReRAM cells. Existing works usually split convolutional computations of one layer to a few pieces, and then assign each piece to one crossbar, which causes a lot of digital-analog signal conversions, transmission, and storage cost. In RFSM, we propose a new switch-matrices-based connection

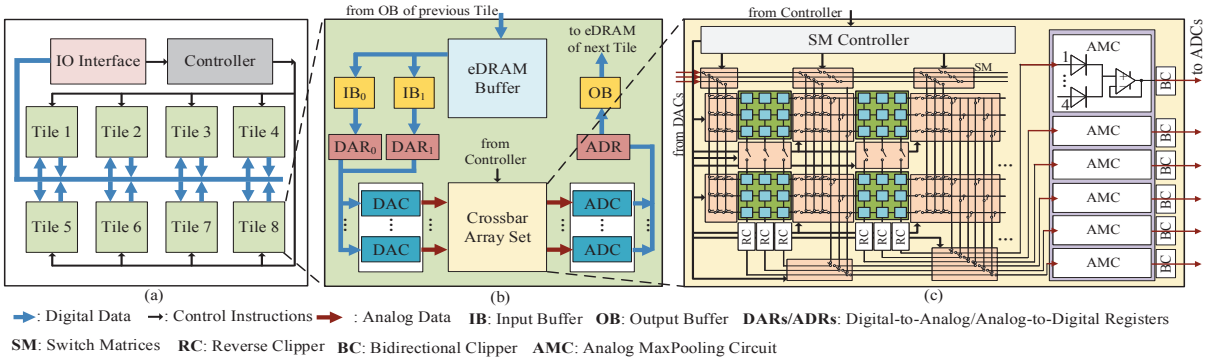


Fig. 2. The RFSM architecture. (a) The structure of RFSM chip. (b) The inter structure of the tile. (c) The structure of crossbar array set.

method, in order to connect multiple crossbars to adapt the demand of convolutional computations, so that the outputs of one convolutional layer could be directly transmitted to the next convolutional layer, and thus reduce the conversions and storage cost. The connections of crossbars are adjusted only once before all convolutional tasks performed. The key steps are as follows.

- The controller sends the parameters such as strides and the center coordinators of the first receptive-field to the switch-matrices (SM) controller.
- The SM controller reconfigures the connection status of the crossbars, by selected a certain number of rows and columns of crossbar for connection. The amount of rows is determined by input size, and the amount of columns is due to the output size. The SM controller could change the switch-matrices status to reconfigure the size of crossbars.
- When the reconfigured crossbars finish the computations, it will send the output signals to other crossbars to do the computations of the next convolutional layer. The mapping is due to the receptive-field-based relationship between the data in adjunct convolutions, which is calculated when the controller has been initialized.

The Pipeline: We design a tile-level pipeline to improve the calculation performance in further. Figure 3 (b) gives the pipeline structure. We assign 6 pipeline stages termed eI, Oe, ID, AO, DD, and AA, as shown in Figure 3 (c). In order to avoid the structural hazard, we set two input buffers and DA registers in a tile. The stage eI_{00} means digital signals are transmitted from eDRAM $buffer_0$ to input $buffer_0$. eI_{01} is performed in the next cycle after eI_{00} completed in one cycle. Oe_1 represents that output buffer sends outputs to eDRAM $buffer_1$ which is in other tile. AO means output digital signals are sent from ADR to output buffer. ID_{00} is for a transmission from input $buffer_0$ to DA $register_0$. DD_0 means digital signals transmission from DA $register_0$ to latches of DACs. AA means a transmission from a latch of a ADC to the AD register. D-C-A indicates the signal conversions and analog convolutions between DAC, crossbar array set and ADC.

Since the inputs could not be sent in only one cycle (usually 100ns), eI, ID and DD may be performed more than one cycle. When all inputs have been sent to DACs, the D-C-A will be performed in the next cycle. Therefore, the RFSM pipeline may be non-linear. We gives a reservation form to illustrate the pipeline working principle, as shown in Figure 3 (d).

IV. Evaluation

A. Experiment Setup

We use CACTI 6.5 [10] at 32 nm to model energy and area for all buffers and on-chip interconnects. The ReRAM-based crossbar array energy and area model are based on research [7]. The power and area models of ADCs and the DACs are cited from the research [11]. We compare our proposed RFSM

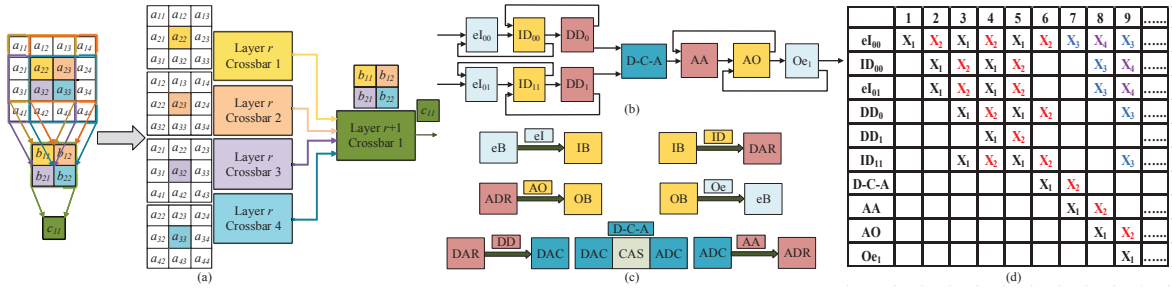


Fig. 3. The diagram of receptive-field based convolution strategy and the RFSM pipeline. (a) The process of 2-layer convolutions. (b) The RFSM pipeline structure. (c) The meaning of each pipeline stage. eB: eDRAM buffer; IB: input buffer; DAR: DA register; CAS: crossbar array set; ADR: AD register; OB: output buffer. (d) The RFSM pipeline reservation form.

TABLE I
PARAMETERS OF RFSM AND ISAAC

Tile Parameters		RFSM	ISAAC
Technology / Frequency		32 nm / 1.2 GHz	
eDRAM Buffer	Size	50 KB	64 KB
	Number	1	1
DAC	Resolution	8 bits	1 bits
	Number	108/18432	12288
ADC	Resolution	8 bits	
	Number	64∼4096	96
ReRAM Crossbars	Size	128 × 128	
	Bits per cell	8 bits	2 bits
	Number	20/1440/2880	96
The number of tiles in one chip		16	168

with the state-of-the-art ReRAM-based accelerator (ISAAC [1]) on the same simulator. The important parameters of RFSM and ISAAC are shown in Table I.

B. The Result and Analysis

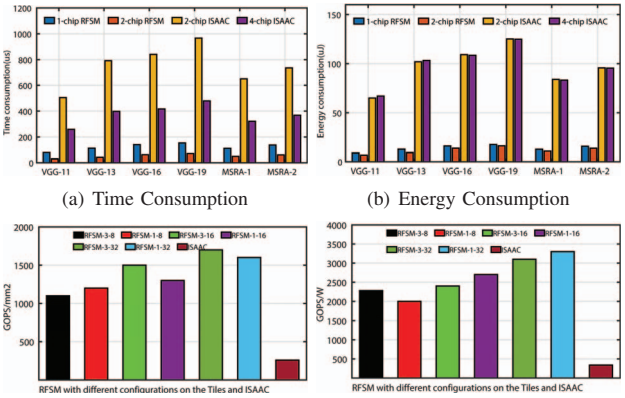
A RFSM chip (20/1440/2880) has 3 types of crossbar array sets with different number of crossbars, in which 20 means that a type of crossbar array set has 20 fixed-size crossbars. Since the inputs for all CNN usually have 3 channels which are much smaller than the channels of intermediate data, the tile with 20 crossbars in crossbar array set is devoted to processing inputs of the first sub-network. The tiles with 1440 or 2880 crossbars are assigned to process the other sub-networks.

We use RFSM chip and ISAAC chip to run CNN inference tasks, and compare their time consumption and energy consumption, respectively. The energy consumption of controller calculating the center coordinators, size and stride is ignored, because the cost of these tasks are very small, and they are executed only once for one certain CNN before all tasks are performed. Two ISAAC chips computation units (crossbars) approximately equal to one RFSM chip (20/1440/2880). Figure 4 (a) and (b) show that compared with two ISAAC chips, one RFSM chip (20/1440/2880) can achieve up to 6.7x speedup and 7.1x lower energy consumption than two ISAAC chips.

We also provide another configuration that all crossbar array sets have 2880 crossbars termed as RFSM chip (2880). The area of a RFSM chip (20/1440/2880) and RFSM chip (2880) are about 183.9 mm^2 and 212.6 mm^2, which are much larger than the area of a ISAAC chip. We compare 2 types of RFSM chips with ISAAC chip in computational efficiency and power efficiency with different number of tiles, and show the results in Figure 4 (c) and (d). Compared to than ISAAC chip, RFSM chip (20/1440/2880) and RFSM chip (2880) provide 7.2x and 6.6x higher computational efficiency, and 9.1x and 9.7x higher power efficiency, respectively.

V. CONCLUSION

Digital-analog conversions exhaust a lot of computation resource and energy resources in PIM-based CNN accelerators. In this work, we first analyze the relationship of input and output data among different convolutional layers based on receptive-field, in order to assure the computation correctness for transmitting the data with analog signals directly. We then design a combination scheme to dynamically reconfigure the size of crossbars, in order to transmit the analog data among convolutions. Our finding illustrates that reducing the number of digital-analog conversions will bring huge performance improvements.



(a) Time Consumption (b) Energy Consumption

(c) Computational Efficiency (d) Power Efficiency

Fig. 4. Time and energy consumption of RFSM and ISAAC. RFSM-3-8 means a RFSM chip (20/1440/2880) with 8 tiles.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (61702013, 62072381), the National Key Research and Development Program of China (2018YFB1800302), the Beijing Natural Science Foundation (L192021, 4202020, KZ201810009011), the Natural Science Foundation of Fujian Province of China (2020J01002), and CCF-Tencent Open Fund WeBank Special Fund.

REFERENCES

[1] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars." ACM SIGARCH Computer Architecture News 44.3 (2016): 14-26.

[2] W. Liu, Z. Wang, X. Liu, N. zeng, Y. Liu and F. E. Alsaadi, A survey of deep neural network architectures and their applications. Neurocomputing 234 (2017): 11-26.

[3] S. Karen, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[4] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang and Y. Xie, Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory. ACM SIGARCH Computer Architecture News 44.3 (2016): 27-39.

[5] H. Mao, M. Song, T. Li, Y. Dai and J. Shu, "Lergan: A zero-free, low data movement and pim-based gan architecture." 2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2018.

[6] N. P. Jouppi, et al., "In-datacenter performance analysis of a tensor processing unit." Proceedings of the 44th Annual International Symposium on Computer Architecture. 2017.

[7] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang and R. S. Williams, "Dot-product engine for neuromorphic computing: programming 1t1m crossbar to accelerate matrix-vector multiplication." 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC). IEEE, 2016.

[8] W. A. Wulf and S. A. McKee, "Hitting the memory wall: implications of the obvious." ACM SIGARCH computer architecture news 23.1 (1995): 20-24.

[9] D. B. Strukov, G. S. Snider, D. R. Stewart, Williams R S, "The missing memristor found." nature 453.7191 (2008): 80-83.

[10] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, "Optimizing nuca organizations and wiring alternatives for large caches with cacti 6.0," in Proceedings of MICRO, 2007.

[11] L. Kull, T. Toifl, M. Schmatz, P. A. Francese, C. Menolfi, M. Brandli, M. Kossel, T. Morf, T. M. Andersen, and Y. Leblebici, A 3.1 mW 8b 1.2 GS/s single-channel asynchronous sar adc with alternate comparators for enhanced speed in 32 nm Digital SOI CMOS, Journal of Solid-State Circuits, 2013.

[12] Hien D H T. A guide to receptive field arithmetic for convolutional neural networks. medium. com, 2017.