Digital-based Processing In-Memory: A Highly-Parallel Accelerator for Data Intensive Applications

Mohsen Imani, Saransh Gupta, and Tajana Rosing

Department of Computer Science and Engineering, UC San Diego, La Jolla, CA 92093, USA
{moimani, sgupta, tajana}@ucsd.edu

ABSTRACT

Recently, Processing In-Memory (PIM) has been shown as a promising solution to address data movement issue in the current processors. However, today's PIM technologies are mostly analog-based, which involve both scalability and efficiency issues. In this paper, we propose a novel digital-based PIM which accelerates fundamental operations and diverse data analytic procedures using processing in-memory technology. Instead of sending a large amount of data to the processing cores for computation, our design performs a large part of computation tasks inside the memory; thus the application performance can be accelerated significantly by avoiding the memory access bottleneck. Digital-based PIM supports bit-wise operations between two selected bit-line of the memory block and then extends it to support row-parallel arithmetic operations.

CCS CONCEPTS

• **Computer systems organization** → **Architectures**; • **Hardware** → **Emerging technologies**.

KEYWORDS

Processing in-memory, Non-volatile memory, Machine learning acceleration

ACM Reference Format:

Mohsen Imani, Saransh Gupta, and Tajana Rosing. 2019. Digital-based Processing In-Memory: A Highly-Parallel Accelerator for Data Intensive Applications. In *Proceedings of the International Symposium on Memory Systems (MEMSYS '19)*, September 30-October 3, 2019, Washington, DC, USA. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3357526.3357551>

1 INTRODUCTION

Today IoT applications analyze raw data by running machine learning algorithms in data centers. However, running data-intensive workloads with large datasets on traditional cores results in high energy consumption and slow processing speed due to a large amount of data movement between memory and processing units. Although new processor technology has evolved to serve computationally complex tasks more efficiently, data movement costs between processor and memory still hinder the higher efficiency of application performance. In addition, applications in this area involve diverse

data analytic procedures which need to be significantly accelerated for handling a large amount of data.

2 PROCESSING IN-MEMORY

The capability of non-volatile memories (NVMs) to act as both storage and a processing unit has encouraged research in processing in-memory (PIM) [1–5]. There are several approaches tried to enable PIM functionality on NVMs, in particular memristor devices. Here, we first show the overview of the commonly used PIM technology supporting dot product operations in the analog domain. Then, we introduce our proposed digital-based PIM technology addressing the several issues of the current PIM technology.

2.1 Analog-based PIM

Vector-matrix multiplication is a popular operation involve in many learning and big data processing applications. The analog-based processing in memory (Analog-PIM) exploits the resistive characteristics of memristor devices in order to enable fast dot product operation. Figure 1a shows the structure of Analog-PIM exploiting an array of a crossbar memory to store matrix values. Each memory element in crossbar memory is a multi-bit memristor device. Instead of computing in the digital domain, Analog-PIM converts input vector to an analog voltage and passes it to a crossbar array. Depending on the converted analog voltage, a current passes through matrix devices enabling analog multiplication of the voltage with the $1/R$, where R is memristor values. Several existing works exploited Analog-PIM to accelerate big data processing applications, such as deep neural network [1, 3], and graph processing [6].

Unfortunately, there are several existing challenges with the current analog-PIM technology: (i) it uses analog-to-digital and digital-to-analog converters (ADCs/DACs) to transfer input data to the analog domain and convert the dot product result back to the digital domain. ADC/DAC blocks are taking the majority of the chip power and area (~98% of the area in DNN accelerator [3]). In addition, since ADC/DAC blocks are mixed-signal, they do not scale as fast as digital technology does. (ii) Analog-PIM uses multi-bit memristor devices to store the matrix values. However, these technologies are not sufficiently reliable. (iii) Analog-PIM computation requires a large amount of internal data movement. This eliminates in-place computation; making the data movement a computation bottleneck on data-intensive workloads, e.g., DNN training.

2.2 Digital-based PIM

Overview: In this paper, we present the idea of digital-based Processing In-Memory, called Digital-PIM, to address the existing challenges with Analog-PIM technology. Digital-PIM performs all essential operations on the digital data stored in a crossbar memory. Digital-PIM supports a row-parallel bitwise NOR operation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MEMSYS '19, September 30-October 3, 2019, Washington, DC, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-7206-0/19/09...\$15.00

<https://doi.org/10.1145/3357526.3357551>

Figure 1: Analog-PIM and Digital-PIM functionalities.

between two selected columns/bitlines of the crossbar memory. This operation is supported internally in memory without any read or data access to the sense amplifier. Digital-PIM extends a series of the NOR-based operations to support arithmetic (addition/multiplication) in a row-parallel way [2, 7–10]. Digital-PIM performs computation on a digital data stored in a crossbar memory; thus removes the necessity of ADC/DAC blocks. It also uses single-bit memristor devices which are compatible with the existing Intel 3D XPoint technology [11]. Finally, Digital-PIM performs in-place computation where the data is stored. Therefore, it eliminates the internal data movement.

Digital-PIM Operations: Instead of working in the analog area, digital-based processing in memory (Digital-PIM) performs the computation directly on the stored values in the memory. Digital-PIM exploits the switching characteristic of memristor devices to internally perform the bitwise computation on the selected memory element without reading them out or using any sense amplifier. Digital-PIM has been designed in literature [8, 12, 13] and fabricated in [14], to implement logic using memristor switching.

Figure 1b shows the structure of Digital-PIM. Digital-PIM exploits crossbar memory with single-bit NVM device and implements NOR operation in a row-parallel way among the selected memory columns. In crossbar, each memristor device switches between two resistive states, R_{ON} (low resistive state, ‘1’) and R_{OFF} (high resistive state, ‘0’), whenever the voltage across the device exceeds a threshold [15]. This property can be exploited to implement NOR gate between the memory elements [12]. Figure 1b also shows the NOR functionality on a single row of a crossbar memory. The To execute NOR in a row, an execution voltage, V_0 , is applied at the p terminals of the inputs devices while the p terminal of the output memristor is grounded. If one or more input memristors are in low resistance state (storing “1” value), the voltage across output device will be V_0 , resulting in switching the output device to high resistance stage (“0” value). However, if all input devices are in high resistance stage, the voltage across the output device cannot switch the output device; thus output device keeps “1” value.

Since NOR is a universal logic gate, it can be used to implement other logic operations like addition [7, 16] and multiplication [17, 18]. Digital-PIM arithmetic operations are in general slower than the corresponding CMOS-based implementations. This is because memristor devices are slow in switching. However, this

Figure 2: Digital-PIM Overview: DNN acceleration [2]

PIM architecture can provide significant speedup with large parallelism. PIM can support addition and multiplications in parallel, irrespective of the number of rows. For example, to add values stored in different columns of memory, it takes the same amount of time for PIM to process the addition in a single row or all memory rows. However, the processing time in conventional cores highly depends on the data size.

3 DEEP LEARNING ACCELERATORS

Several existing works already show how to exploit Digital-PIM for the acceleration of deep learning and big data processing applications [2, 7, 19]. Here, we explain how Digital-PIM can be used to accelerate Deep Neural Networks (DNNs) inference and training phases.

Figure 2a shows an overview of the Digital-PIM architecture consisting of multiple crossbar memory blocks. As an example, Figure 2b shows how three adjacent layers are mapped to Digital-PIM memory blocks to perform the feed-forward computation. Each memory block represents a layer of the neural network, and stores the data used in either testing (i.e., weights) or training (i.e., weights, the output of each neuron before activation, and the derivative of the activation function (g')), as shown in Figure 2c. With the stored data, the Digital-PIM performs with two phases: (i) computing phase and (ii) data transfer phase. During the computing phase, all memory blocks work in parallel, where each block processes an individual layer using PIM operations. Then, in the data transfer phase, the memory blocks transfer their outputs to the blocks corresponding to the next layers, i.e., to proceed either the feed-forward or back-propagation. The switches are shown in Figure 2b control the data transfer flows.

We present how each memory block performs DNN computations for a layer. The block supports in-memory operations for key CNN computations, including vector-matrix multiplication, convolution, and pooling. We also support the activation functions like ReLU and Sigmoid in memory. MIN/MAX pooling operations are implemented using in-memory search operations. Our proposed design optimizes each of the basic operations to provide high performance. The feed-forward step is performed entirely inside memory

by executing the basic PIM operations. We also perform all the computations of the back-propagation with the same key operations and hardware to the one used in the feed-forward.

We evaluate the efficiency of on ImageNet dataset using popular large-scale neural networks. Thanks to bitwise computation capability, Digital-PIM is the first PIM architecture that natively supports floating-point computation [2]. Floating-point precision is essential for DNN training. Our evaluation shows that Digital-PIM can achieve up to 5.1% higher classification accuracy as compared to existing PIM architectures with limited fixed-point precision. In addition, Digital-PIM training is on average $303.2\times$ and $48.6\times$ ($4.3\times$ and $15.8\times$) faster and more energy efficient as compared to GTX 1080 GPU (PipeLayer [20], Analog-PIM accelerator).

ACKNOWLEDGEMENTS

This work was partially supported by CRISP, one of six centers in JUMP, an SRC program sponsored by DARPA, and NSF grants 1527034, 1730158, 1826967, 1911095.

REFERENCES

- [1] 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," in *Proceedings of the 43rd International Symposium on Computer Architecture*, pp. 27–39, IEEE Press, 2016.
- [2] M. Imani, S. Gupta, Y. Kim, and T. Rosing, "Floatpim: In-memory acceleration of deep neural network training with high precision," in *IEEE/ACM International Symposium on Computer Architecture (ISCA)*, 2019.
- [3] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramanian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in cross-bars," in *Proceedings of the 43rd International Symposium on Computer Architecture*, pp. 14–26, IEEE Press, 2016.
- [4] M. Imani, M. Samragh, Y. Kim, S. Gupta, F. Koushanfar, and T. Rosing, "Rapidnn: In-memory deep neural network acceleration framework," *arXiv preprint arXiv:1806.05794*, 2018.
- [5] M. Imani, A. Rahimi, D. Kong, T. Rosing, and J. M. Rabaey, "Exploring hyperdimensional associative memory," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 445–456, IEEE, 2017.
- [6] L. Song, Y. Zhuo, X. Qian, H. Li, and Y. Chen, "Graphr: Accelerating graph processing using reram," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 531–543, IEEE, 2018.
- [7] M. Imani, S. Gupta, and T. Rosing, "Ultra-efficient processing in-memory for data intensive applications," in *Proceedings of the 54th Annual Design Automation Conference 2017*, p. 6, ACM, 2017.
- [8] S. Gupta *et al.*, "Felix: Fast and energy-efficient logic in memory," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–7, IEEE, 2018.
- [9] M. Imani *et al.*, "Digitalpim: Digital-based processing in-memory for big data acceleration," in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, pp. 429–434, ACM, 2019.
- [10] M. Zhou *et al.*, "Thermal-aware design and management for search-based in-memory acceleration," in *Proceedings of the 56th Annual Design Automation Conference 2019*, p. 174, ACM, 2019.
- [11] "Intel and micron produce breakthrough memory technology.." http://newsroom.intel.com/community/intel_newsroom/blog/2015/07/28/intel-and-micron-produce-breakthrough-memory-technology.
- [12] S. Kvatinisky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Magic—memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, 2014.
- [13] S. Salamat, M. Imani, S. Gupta, and T. Rosing, "Rnsnet: In-memory neural network acceleration using residue number system," in *2018 IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1–12, IEEE, 2018.
- [14] B. C. Jang, Y. Nam, B. J. Koo, J. Choi, S. G. Im, S.-H. K. Park, and S.-Y. Choi, "Memristive logic-in-memory integrated circuits for energy-efficient flexible electronics," *Advanced Functional Materials*, vol. 28, no. 2, 2018.
- [15] S. Kvatinisky, M. Ramadan, E. G. Friedman, and A. Kolodny, "Vteam: A general model for voltage-controlled memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786–790, 2015.
- [16] N. Talati, S. Gupta, P. Mane, and S. Kvatinisky, "Logic design within memristive memories using memristor-aided logic (magic)," *IEEE Transactions on Nanotechnology*, vol. 15, no. 4, pp. 635–650, 2016.
- [17] A. Haj-Ali *et al.*, "Efficient algorithms for in-memory fixed point multiplication using magic," in *IEEE ISCAS*, IEEE, 2018.
- [18] S. Gupta *et al.*, "Exploring processing in-memory for different technologies," in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, pp. 201–206, ACM, 2019.
- [19] M. Zhou *et al.*, "Gram: graph processing in a reram-based computational memory," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pp. 591–596, ACM, 2019.
- [20] L. Song, X. Qian, H. Li, and Y. Chen, "Pipelayer: A pipelined reram-based accelerator for deep learning," in *High Performance Computer Architecture (HPCA), 2017 IEEE International Symposium on*, IEEE, 2017.