UPIM : Unipolar Switching Logic for High Density Processing-in-Memory Applications

Joonseop Sim*, Saransh Gupta*, Mohsen Imani, Yeseong Kim, and Tajana Rosing
University of California San Diego
{j7sim,sgupta,moimani,yek048,tajana}@ucsd.edu

ABSTRACT

Internet of Things (IoT) has built a network with billions of connected devices which generate massive volumes of data. Processing large data on existing systems requires significant costs for data movements between processors and memory due to limited cache capacity and memory bandwidth. Processing-In-Memory (PIM) is a promising solution to address the issue. Prior techniques that enable the computation in non-volatile memory (NVM) are designed on a bipolar switching mode, which suffers from a high sneak current in a crossbar array (CBA) structure. In this paper, we propose a unipolar-switching logic for high-density PIM applications, called UPIM. Our design exploits a unipolar-switching mode of memristor devices which can be operated in 1D1R structure hence suppresses the sneak current that exists in prior PIM technologies. Moreover, UPIM takes advantages of a 3D vertical crossbar array (CBA) structure to increase memory utilization per unit area for high-density applications. Our evaluation on a wide range of applications shows that the UPIM achieves up to 31.3× energy saving and 113.8× energy-delay product (EDP) improvement as compared to a recent GPGPU architecture. As compared to the state-of-the-art PIM design based on the bipolar switching mode, our design achieves 3.1× lower energy consumption.

KEYWORDS

PIM; Memristor; Sneak current; Unipolar switching; 3D CBA

ACM Reference Format:
Joonseop Sim*, Saransh Gupta*, Mohsen Imani, Yeseong Kim, and Tajana Rosing. 2019. UPIM : Unipolar Switching Logic for High Density Processing-in-Memory Applications. In *Great Lakes Symposium on VLSI 2019 (GLSVLSI '19), May 9–11, 2019, Tysons Corner, VA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3299874.3318011

1 INTRODUCTION

Many of today's applications face huge challenges to improve their performance, since the amount of data to be processed significantly increases with the emergence of the Internet of Things [1]. Although electronic circuit integration and scaling of semiconductor devices have significantly increased the number of processing units and memory sizes, the limited on-chip memory capacity and memory bandwidth hinder further efficiency improvement on the conventional computing systems [2–4]. Processing-in-memory (PIM) is a promising technique to address the issue of data movement congestion between processing cores and memory by performing essential operations inside the memory, instead of sending all the data to processing cores [5–11]. It can significantly reduce the amount of data transferred between the memory and processing cores, thus accelerating big data applications.

Prior studies show that non-volatile memories (NVMs) have great potential to enable the PIM functionalities by exploiting analog characteristics of the memory devices. High density, lower energy consumption, and scalability of NVMs make them suitable candidates to replace the memories in conventional architectures [12]. For example, the work in [13, 14] proposed a modified sense amplifier to enable bitwise operations. The work in [15–18] support in-memory bitwise operations or arithmetic operations with a few Boolean functions. The existing NVM-based PIM designs enable essential logic functions in a crossbar array (CBA) structure which is suitable for memristor devices due to its small cell size ($4F^2$). However, when applying their techniques to a large CBA structure, there are two main drawbacks. First, most of these designs work with a bipolar switching mode, which is vulnerable to sneak current paths during read and write operations [19]. The sneak current becomes more serious when the array size increases because it increases the number of unselected cells which form undesired sneak current paths. Second, to execute arithmetic operations, e.g., addition and multiplication, they require extra cells to store intermediate computation results with multiple cycles of bitwise operations [6, 18, 20]. This hinders area efficiency for the high-density applications.

In this paper, we propose *UPIM*, a novel processing-in-memory architecture, which enables PIM functions using unipolar-switching mode with 3D-layered structure. The main contributions of this paper include:

- To the best of our knowledge, this is the first work that supports PIM logic with unipolar-switching memristors for the 1D1R cell structure. This design offers a sneak current reduction compared to the existing bipolar-based 1R structure for a high-density CBA.
- We show our proposed design that supports fundamental Boolean operations in the memory, including NOR, NAND, and NOT. They enable PIM-based arithmetic operations, e.g., addition and multiplication.
- We also propose a PIM-enabled 3D vertical crossbar structure to further increase the area efficiency by overlapping the intermediate cells.

Our experimental results show that the proposed design achieves up to 31.3× energy saving and 113.8× EDP improvement, as compared to a recent GPGPU architecture.

2 DESIGN OVERVIEW

2.1 Memristor Switching Modes

There are two classes of ReRAM switching mode depending on the applied bias polarity. One is 'unipolar', where the switching between high resistance state (HRS) and low resistance state (LRS) is not relevant to the polarity of the operating voltage and the other is 'bipolar', where the reset switching (LRS → HRS) and set

*Joonseop Sim and Saransh Gupta contributed equally to this research.

GLSVLSI '19, May 9–11, 2019, Tysons Corner, VA, USA

ACM ISBN 978-1-4503-6252-8/19/05…$15.00
https://doi.org/10.1145/3299874.3318011

switching (HRS → LRS) take place with the opposite of the bias polarity [21]. Unipolar switching has the following advantages: (i) the symmetric property in polarity which provides an easier implementation in the memory arrays and (ii) reducing the sneak current and write disturb by adding a selector device such as a diode [22]. Consider an M×N array, there are $(N-1)(M-1)$ sneak current paths when a single cell on the black line is intended to be read. Therefore, the total current includes the summation of sneak current with original cell current. The overall sneak current can be represented by Eq. (1) [19].

$$I_{SNEAK} = V_R \times \left(\frac{R_F}{N-1} + \frac{R_R}{NM-M-N+1} + \frac{R_F}{M-1} \right)^{-1} \quad (1)$$

where V_R is the applied voltage, and R_F and R_R are the corresponding resistance when *forward* and *reverse* current flow, respectively. Eq. (1) has a significant implication that since majority of cells have sneak current paths in reverse direction, increasing R_R to rectify the reverse current is a critical requirement to suppress sneak current dissipation. In contrast to most prior work which enables PIM functions in bipolar devices [6, 15, 16, 18], in this paper we propose a unipolar-based logic family which can reduce the sneak current and results in static energy saving. In the following subsections, we explain how the design enables logic functions using unipolar devices.

2.2 Unipolar-based logic within NVM

Fig. 1(a) shows the basic structure of the proposed UPIM. To simplify the explanation, we show a logic that supports two-input NOR operation, but it can be extended to multi-input logics in a straight-forward way. Each unipolar device consists of a memristor device and a diode. The input values are stored in two memristors, R_{IN1} and R_{IN2}, while the other memresistor, R_{OUT} stores the computation result. The logical values are stored in each memresistor as resistance states in the input/output memristors. HRS in either $R_{IN1,2}$ or R_{OUT} indicates the logical value of 0, while LRS represents 1. In our experiment, we exploit the memristor model in [23], whose R_{LRS} and R_{HRS} are 10KΩ and 10MΩ, respectively. Our logic also has one additional resistor, R_G, whose resistance is configurable. In this work, we select $R_G = 300K\Omega$, a value between R_{LRS} and R_{HRS} based on the consideration of process variation. We explain the detailed configuration in Section 3.3. All four resistors are connected to the BL. Fig. 1(b) shows how to set the operation voltage, V_{IN} and V_{OUT}, considering V_{SET}. In our design, V_{IN} has a lower voltage than V_{SET}, and a V_{OUT} is higher than V_{SET}.

To perform the NOR operation, our design first initializes the R_{OUT} to R_{HRS}. We then apply the V_{IN1} and V_{IN2} voltages to the input memristors and V_{OUT} to the output memristor. Fig. 1(c) shows how the proposed logic performs the NOR operation. In the two-input case, the stored values in the input memristors have four combinations: 00, 01, 10 and 11. When both inputs have high resistance, i.e., '00', the voltage on the BL (V_{BL}) is almost pulled into ground, while the voltage across R_{OUT} (V_{OUT}-V_{BL}) is close to V_{OUT}. Since V_{OUT}-V_{BL} is larger than V_{SET}, it incurs the SET switching of the R_{OUT} to R_{LRS}. Note that the applied voltage across the diode is negligible as compared to the voltage applied to R_{OUT} since R_{OUT} is previously initialized as HRS. In all other cases (i.e., 01, 10, and 11), at least one of the input memristors has a low resistance state. Therefore, the V_{BL} voltage has a higher voltage close to V_{IN}. For instance, if the case of '10', where R_{IN1} and R_{IN2} have LRS and HRS, respectively, the net resistance is close to R_{IN1}. Since the voltage ratio of R_{IN1} to R_G is close to zero, (≈ 0.03 in our experiment), V_{BL} is almost V_{IN}. Thus, the R_{OUT}

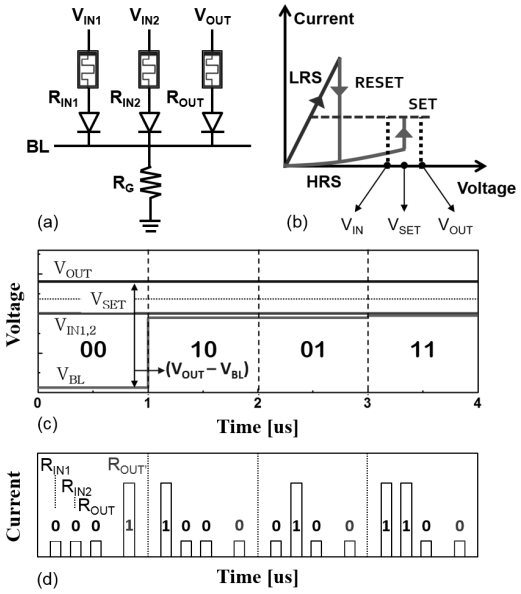


Figure 1: Proposed unipolar–based NOR logic: (a) Schematic of the NOR gate (b) Voltage conditions (c) NOR gate simulation result (d) Resistance behaviors depending on input states

keeps the high resistance state representing the logical 0. Fig. 1(d) shows the resistance behavior of the UPIM NOR gate. R_{OUT} and R_{OUT}, indicate resistance states from the output resistor prior to operation and after applying V_{IN}, respectively. Except for the case of '00' which the SET switching occurs in R_{OUT}, all the other cases keep the R_{OUT} as high resistance state, presenting NOR operation.

2.3 Integration to 3D CBA structure

The proposed design executes arithmetic functions using NOR operations. Existing NOR-based approaches require additional cells to store intermediate results. The area overhead due to the generated intermediate states is not suitable for high-density applications. In this work, we utilize a 3D structure to minimize the area cost. Fig. 2(a) shows the conventional 2D logic implemented in a memory array. In this structure, the intermediate operation results are stored in the same plane while consuming an extra cell area. In contrast, as shown in Fig. 2(b), the 3D structure can store the intermediate results in a different layer. Therefore, the intermediate cell is hidden under/over the memory cells, increasing chip density as compared to the 2D case.

Fig. 3 presents the comparison diagram of 2D and 3D cases. We denote the area of memory cells, which is used to store data, by A_{memory}. A_{logic} and A_{shift} are the areas of intermediate cells for storing logic results and of the interconnects, respectively. We define *cell efficiency* as the ratio of the memory area over the total area. In the 2D design, since the intermediate cells take chip area, the cell efficiency is represented by $A_{memory}/(A_{memory} + A_{logic} + A_{shift})$. In contrast, for the 3D case, the intermediate cells for all arithmetic logic can be completely stacked on the top of the memory cells. If the number of layers is n, the cell efficiency of 3D design is given by $(n \times A_{memory})/(A_{memory} + A_{shift})$. This means that, with the 3D logic stacking, it can achieve high area efficiency.

Fig. 4 shows our integration design of 3D logic-in-memory. The V_{IN} and V_{OUT} are applied to wordlines connected to memory cells and intermediate cells, respectively. For example, if the V_{IN} is applied to 'A' and 'B' cell, the result of NOR operation is stored at a cell where the V_{OUT} is applied. As appeared in the figure, the

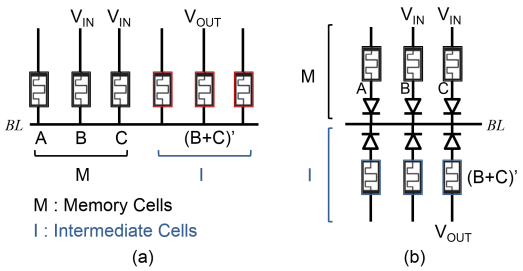


Figure 2: Schematic of (a) Prior 2D and (b) Proposed 3D logic in memory

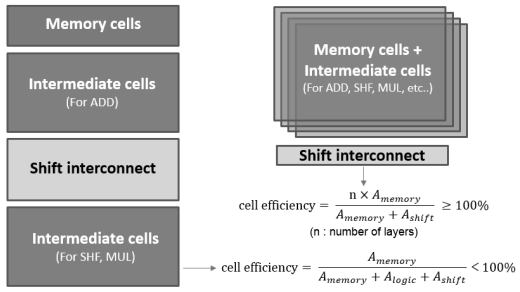


Figure 3: Diagram of prior 2D (left) and proposed 3D (right) PIM structures

proposed 3D structure can improve the chip density by storing the intermediate results in a different layer compared to the 2D structure. Moreover, a memory layer and a computation layer are paired and they can be stacked with multiple layers. Therefore, our design enables parallel operation with a single input signal. In the case of Fig. 4, the UPIM NOR operations of A and B, D and E can be executed in parallel with a single PIM operation. Table. 1 summarizes the comparison of the proposed UPIM to existing technologies.

3 EXPERIMENTAL RESULTS

3.1 Experimental Setup

Performance and energy consumption have been obtained by Cadence Virtuoso and Spectre simulators with 45nm CMOS process technology. We use *VTEAM* memristor model [23] with R_{LRS} and R_{HRS} of 10KΩ and 10MΩ respectively. We implement the diode model with saturation current (*IS*), ohmic resistance (*RS*) and emission coefficient (*N*) for 1.8e-5*A*, 1.43Ω and 1.22, respectively. We compare the efficiency of the proposed UPIM design with AMD R390 GPU and state-of-the-art PIM designs, [15, 18, 24].

3.2 Energy and Performance

As discussed in Section 2.1 and 2.2, our unipolar-based logic is operated in the 1D1R structure, which shows lower static power consumption by reducing sneak current dissipation. Fig. 5 shows the energy and energy-delay product (EDP) improvements of running applications on proposed UPIM and state-of-the-art PIM designs [15, 18, 24], which use 1D1R and 1R cell structures, respectively. All results are normalized to energy and EDP of AMD GPU. For each application, the size of the input dataset is fixed to 512MB. In traditional cores, the energy and performance of computation consist of two terms: computation and data movement. In conventional cores, the data movement is restricted by a small cache size of a transitional core which increases the number of cache miss. Consecutively, this degrades the energy consumption and performance of data movement between the memory and caches. In contrast, in PIM architecture the dataset is already stored in the memory and computation is a major cost. Although the memory-based computation is slower than transitional CMOS-based computation (*i.e.* floating point units in GPU), in processing the large dataset, the

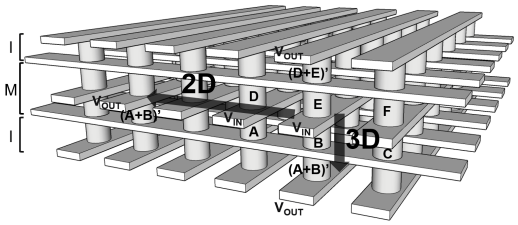


Figure 4: The integrated structure of 3D UPIM

Table 1: performance of proposed UPIM and other technologies

	IMPLY [18]	MAGIC [15]	2D-UPIM	3D-UPIM
Cell Structure	1R	1R	1D1R	1D1R
Condition	$2(V_{cond}, V_{set})$	$1(V_0)$	$2(V_{in}, V_{out})$	$2(V_{in}, V_{out})$
Functions	IMPLY (False)	OR, NOR, NOT, AND, NAND		
Power	High leakage	High leakage	**Low**	**Low**
Density	Low	Low	Low	**High**

PIM works significantly faster than GPU. Our evaluation shows that UPIM achieves 31.3× energy efficiency, and 113.8× EDP improvement as compared to GPU. As compared to the state-of-the-art PIM design based on the bipolar switching mode, UPIM achieves 3.1× lower energy consumption. The higher efficiency of the UPIM comes from its more efficient approach in calculating a single NOR operation as explained in Sec. 2.2.

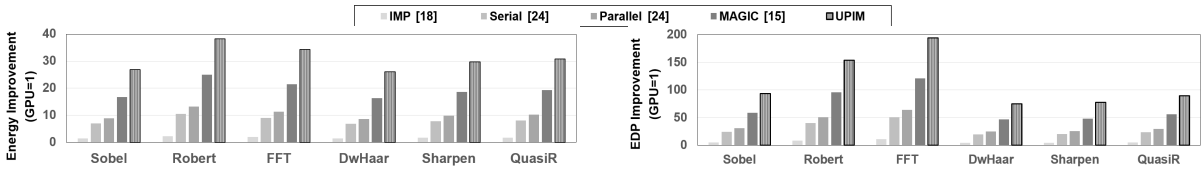
3.3 Process Variation

The UPIM design uses a configurable resistor, R_G. To make our design robust, we determine the resistor value with consideration of process variation, which most of today's technology suffers. In our experiment, there are two major factors that induce process variation, memristor dimension, and near-far cell difference. The dimension variation comes from a diameter deviation during lithograph and etching process in the formation of pillar memristors, which results in the resistance variation on UPIM [25]. The resistance variation also occurs between near and far cells in a memory array. We consider the *near-far effect* on the UPIM operations in a mat array with the size of 1Mb as shown in Fig. 6(a). When R_G and R_{OUT} are located on an edge of the mat, the resistance recognized from either R_G or R_{OUT} are different over near and far cells. For example, in the case of the far cells, the BL resistance are added to a memristor resistance. Since V_{BL} is the electrical potential of the point at which R_G meets the BL, $R_{IN}[1023]$ additionally includes the resistance of the BL connected to 1024 cells, while $R_{IN}[0]$ does not have such an effect.

Fig. 6(b) shows V_{BL} characteristic as a function of R_G, when input values are 00 and 01, considering the factors of the process variation. All V_{BL} transfer curves are presented with dimension variation of 10%, denoted as (H). As R_G increases, the electrical potential in the BL increases due to an escalation of the voltage applied to R_G. $V_{OUT} - V_{BL}$ has to be higher than V_{SET} for the case of 00 and lower than V_{SET} for other cases, i.e., 10,01,11. Thus, the gap between $V_{OUT} - V_{BL}$@10 and $V_{OUT} - V_{BL}$@00 needs to be enough wide for operation stability. The voltage gap, denoted as V_{BL} margin, is tunable by adjusting R_G value. Fig. 6(c) shows the simulation results of the V_{BL} margin for different R_G. We extract an optimized R_G point from the graph of V_{BL} margin with an R_G. Based on this analysis, we choose the optimal R_G value, $R_{G,OPT}$, by 300KΩ to guarantee computation accuracy, despite existing process instability.

3.4 Evaluation for Area Efficiency

We evaluated the area efficiency of our design as compared to the MAGIC [15], a state-of-the-art PIM design. The area efficiency of the PIM techniques is mainly dependent on two factors, i.e., the



(a) Energy (b) Energy-Delay Product

Figure 5: Energy and Energy-delay product improvement of UPIM and state-of-the-art [15, 18, 24] for different applications.

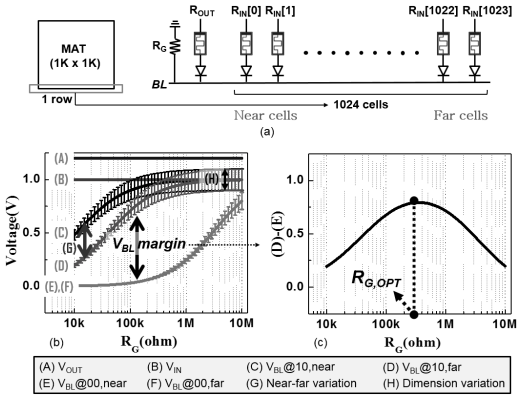


Figure 6: V_{BL} margin and R_G optimization

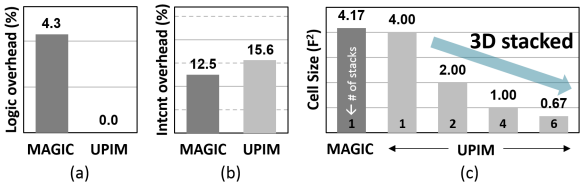


Figure 7: Overhead and cell size comparison between UPIM and MAGIC [15]

area overhead for intermediate cells and for interconnects. The two overheads were defined by A_{LOGIC} / A_{MEM} and A_{INT} / A_{MEM} where A_{LOGIC} and A_{INT} are the area of additional cells for logic functions and interconnect design, and A_{MEM} is the area of original memory cells. As shown in Fig 7(a), the UPIM outperforms the MAGIC in terms of the logic overhead. Since the UPIM design stacks the intermediate cells on different layers, it can implement the arithmetic operations, i.e., addition and multiplications, without area penalty for the logic. On the other hand, Fig. 7(b) shows the effect of 3D-stacked structure on integration density for the interconnects. The result shows that the interconnects in the 3D-stacked design require additional overhead for vertical shifts. However, since the UPIM design exploits the vertical transistors for the interconnects, it occupies less area over the conventional planar transistor. This makes the interconnect overhead minimal, i.e., only 3.1% compared to the MAGIC design. Fig. 7(c) shows the area efficiency comparison in terms of the cell size for different 3D stack decisions. Although the cell size difference between UPIM and MAGIC is less than 5% in a single layer, the efficiency increases as more stacks are exploited. For the six-layered structure, the cell size of UPIM reaches $0.67F^2$, which is much less than $4F^2$, considered as the minimum cell size of 2D-based memristor design.

4 CONCLUSION

We present an energy efficient and high-density PIM architecture which enables logic-in-memory based on unipolar-switching memristors. The proposed design resolves the static power issue due to the sneak current by implementing the logic in the 1D1R cell structure. Our design also addresses the low cell-density of other PIM technologies due to extra area consumption for storing computation results by implementing them in 3D CBA. The experimental results show that our design presents 3.1× and 31.3× improvement in energy consumption compared to the state-of-the-art PIM designs and the GPU architecture, respectively.

ACKNOWLEDGMENTS

This work was partially supported by CRISP, one of six centers in JUMP, an SRC program sponsored by DARPA, and also NSF grants #1730158 and #1527034. Joonseop Sim is partially supported by a Ph.D. fellowship from SK Hynix Inc.

REFERENCES

[1] C. Yang *et al.*, "Big data and cloud computing: innovation opportunities and challenges," *International Journal of Digital Earth*, 2017.
[2] M. Ieong *et al.*, "Silicon device scaling to the sub-10-nm regime," *Science*, 2004.
[3] M. Imani *et al.*, "A framework for collaborative learning in secure high-dimensional space," in *Cloud Computing (CLOUD)*, IEEE, 2019.
[4] G. Dimitroulakos *et al.*, "Alleviating the data memory bandwidth bottleneck in coarse-grained reconfigurable arrays," in *Application-Specific Systems, Architectures Processors, 2005. 16th IEEE International Conference on*, IEEE, 2005.
[5] J. Sim *et al.*, "Lupis: latch-up based ultra efficient processing in-memory system," in *19th International Symposium on Quality Electronic Design (ISQED)*, IEEE, 2018.
[6] S. Gupta *et al.*, "Felix: Fast and energy-efficient logic in memory," in *ICCAD*, pp. 1–7, IEEE, 2018.
[7] M. Imani *et al.*, "Rapidnn: In-memory deep neural network acceleration framework," *arXiv preprint arXiv:1806.05794*, 2018.
[8] M. Zhou *et al.*, "Gas: A heterogeneous memory architecture for graph processing," in *ISLPED*, p. 27, ACM, 2018.
[9] M. Imani *et al.*, "Exploring hyperdimensional associative memory," in *HPCA*, pp. 445–456, IEEE, 2017.
[10] S. Gupta *et al.*, "Nnpim: A processing in-memory architecture for neural network acceleration," *IEEE Transactions on Computers*, pp. 1–1, 2019.
[11] Y. Kim *et al.*, "Orchard: Visual object recognition accelerator based on approximate in-memory processing," in *ICCAD*, pp. 25–32, IEEE, 2017.
[12] L. Chang *et al.*, "Reconfigurable processing in memory architecture based on spin orbit torque," in *2017 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, IEEE, 2017.
[13] S. Li *et al.*, "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in *Design Automation Conference (DAC), 2016 53nd ACM/EDAC/IEEE*, IEEE, 2016.
[14] M. Imani *et al.*, "Mpim: Multi-purpose in-memory processing using configurable resistive memory," in *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*, pp. 757–763, IEEE, 2017.
[15] S. Kvatinsky *et al.*, "Magic: Memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2014.
[16] M. Imani *et al.*, "Ultra-efficient processing in-memory for data intensive applications," in *the 54th Annual Design Automation Conference 2017*, ACM, 2017.
[17] M. Imani *et al.*, "Floatpim: In-memory acceleration of deep neural network training with high precision," in *ISCA*, ACM, 2019.
[18] E. Lehtonen *et al.*, "Stateful implication logic with memristors," in *Proceedings of the 2009 IEEE/ACM International Symposium on Nanoscale Architectures*, IEEE Computer Society, 2009.
[19] J. Y. Seok *et al.*, "A review of three-dimensional resistive switching cross-bar array memories from the integration and materials property points of view," *Advanced Functional Materials*, 2014.
[20] N. Talati *et al.*, "Logic design within memristive memories using memristor-aided logic (magic)," *IEEE Transactions on Nanotechnology*, 2016.
[21] T.-C. Chang *et al.*, "Resistance random access memory," *Materials Today*, 2016.
[22] E. Amrani *et al.*, "Logic design with unipolar memristors," in *Very Large Scale Integration (VLSI-SoC), 2016 IFIP/IEEE International Conference on*, IEEE, 2016.
[23] S. Kvatinsky *et al.*, "Vteam: A general model for voltage-controlled memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2015.
[24] A. Siemon, S. Menzel, R. Waser, and E. Linn, "A complementary resistive switch-based crossbar array adder," *IEEE journal on emerging and selected topics in circuits and systems*, vol. 5, no. 1, pp. 64–74, 2015.
[25] F. W. Sears *et al.*, *College physics*. Addison Wesley Publishing Company, 1974.